

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2014 年 7 月 10 日(10.07.2014)



(10) 国際公開番号
WO 2014/106883 A1

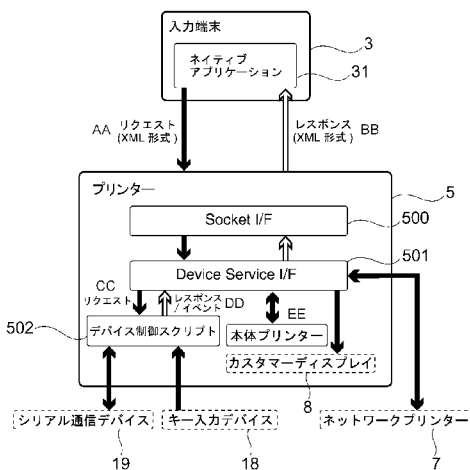
- (51) 国際特許分類:
G06F 13/00 (2006.01)
- (21) 国際出願番号: PCT/JP2013/007633
- (22) 国際出願日: 2013 年 12 月 26 日(26.12.2013)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
61/748,232 2013 年 1 月 2 日(02.01.2013) US
- (71) 出願人: セイコーエプソン株式会社 (SEIKO EPSON CORPORATION) [JP/JP]; 〒1630811 東京都新宿区西新宿二丁目 4 番 1 号 Tokyo (JP).
- (72) 発明者: 高栖 和弘 (TAKASU, Kazuhiro); 〒3928502 長野県諏訪市大和 3 丁目 3 番 5 号セイコーエプソン株式会社内 Nagano (JP). 堤 浩一郎 (TSUTSUMI, Koichiro); 〒3928502 長野県諏訪市大和 3 丁目 3 番 5 号セイコーエプソン株式会社内 Nagano (JP). 池田 茂夫 (IKEDA, Shigeo); 〒3928502 長野県諏訪市大和 3 丁目 3 番 5 号セイコーエプソン株式会社内 Nagano (JP).
- (74) 代理人: 上柳 雅誉, 外 (KAMIYANAGI, Masataka et al.); 〒3990785 長野県塩尻市広丘原新田 8 0 セイコーエプソン株式会社 知的財産本部内 Nagano (JP).
- (81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシ

[続葉有]

(54) Title: DEVICE CONTROL SYSTEM AND METHOD FOR CONTROLLING DEVICE CONTROL SYSTEM

(54) 発明の名称: デバイス制御システム、及び、デバイス制御システムの制御方法

100



(57) Abstract: The present invention enables a device to be controlled by a terminal connected to a network. A device control system is provided with a terminal (3) equipped with an application (31), and a printer (5) for controlling a device. The terminal (3) transmits a request written in an XML format and generated by the application (31). The printer (5) receives the request in XML format, transmits a response in XML format to the terminal (3), and controls the device when the terminal (3) requests that the device be controlled by the request in XML format.

(57) 要約: ネットワーク接続された端末によりデバイスを制御できるようにする。アプリケーション 31 を備えた端末 3 と、デバイスを制御するプリンター 5 とを備えたデバイス制御システムである。端末 3 は、アプリケーション 31 が生成する XML で記述されたリクエストを送信する。プリンター 5 は、XML のリクエストを受信し、端末 3 が XML のレスポンスを送信し、端末 3 が XML のリクエストでデバイスの制御を要求した場合にデバイスを制御する。

- 3 INPUT TERMINAL
- 5 PRINTER
- 7 NETWORK PRINTER
- 8 CUSTOMER DISPLAY
- 18 KEY INPUT DEVICE
- 19 SERIAL COMMUNICATION DEVICE
- 31 NATIVE APPLICATION
- 500 SOCKET INTERFACE
- 501 DEVICE SERVICE INTERFACE
- 502 DEVICE CONTROL SCRIPT
- AA REQUEST (XML FORMAT)
- BB RESPONSE (XML FORMAT)
- CC REQUEST
- DD RESPONSE/EVENT
- EE MAINFRAME PRINTER

WO 2014/106883 A1



ア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ
(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR,
GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT,
NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI
(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML,
MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告 (条約第 21 条(3))

明 細 書

発明の名称：

デバイス制御システム、及び、デバイス制御システムの制御方法

技術分野

[0001] 本発明は、デバイスを制御するデバイス制御システム、及び、デバイス制御システムの制御方法に関する。

背景技術

[0002] 従来、ネットワークに接続された端末により、キーボード、バーコードスキャナー等のデバイスを制御するシステムでは、デバイスの制御用に、コントローラーとなる機器（コンピューターなど）が用意されていた。

先行技術文献

特許文献

[0003] 特許文献1：特開2004-152256号公報

発明の概要

発明が解決しようとする課題

[0004] しかしながら、シンプルな構成で上述したようなデバイスを使用したいというニーズがあった。

本発明は、上述した事情に鑑みてなされたものであり、ネットワーク接続された端末によりデバイスを制御できるデバイス制御システム、端末、及び、デバイス制御システムの制御方法を提供することを目的とする。

課題を解決するための手段

[0005] 上記目的を達成するために、本発明のデバイス制御システムは、データの入力もしくは出力を行うデバイスと、マークアップ言語のリクエストの送信及びマークアップ言語のレスポンスを受信するアプリケーションを実行する端末と、前記デバイスと接続し、前記マークアップ言語のリクエストに基づく前記デバイスの制御及び前記マークアップ言語のレスポンスの送信を行う

デバイス制御装置と、を備えることを特徴とする。

本発明によれば、ネットワーク接続された端末により、デバイス制御装置に接続されたデバイスを制御できる。また、端末は、XMLで記述されたデータを生成するアプリケーションを実行できる装置であれば、制限されない。XMLは汎用性が高いので、様々な装置を端末として使用できる。また、端末のアプリケーションの開発が容易である。

[0006] また、レスポンスは、前記デバイスの制御の結果を示すデータを含む。

また、前記アプリケーションは、前記デバイス制御装置に通信経路の確立を要求する通信確立メッセージを含む前記リクエストを前記デバイス制御装置に送信し、前記デバイス制御装置は、前記通信確立メッセージを含む前記リクエストを受信した時、前記リクエストに含まれる前記通信確立メッセージに対する応答メッセージを含む前記レスポンスを前記アプリケーションに送信する。

[0007] また、前記アプリケーションは、前記デバイス制御装置で送信された前記レスポンスに含まれる前記通信確立メッセージに対する応答メッセージに基づいて前記デバイス制御装置との前記通信経路を確立する。

[0008] また、前記アプリケーションは、前記通信経路が確立した後、前記デバイス制御装置に前記デバイス制御装置に接続された前記デバイスとの通信をオープンするオープンメッセージを含む前記リクエストを前記デバイス制御装置に送信する。

[0009] また、前記デバイス制御装置は、前記オープンメッセージを含む前記リクエストを受信した時、前記リクエストに含まれる前記オープンメッセージに対する応答メッセージを含む前記レスポンスを前記アプリケーションに送信する。

また、前記アプリケーションは、前記デバイス制御装置で送信した前記オープンメッセージに対する応答メッセージを含む前記レスポンスを受信し、前記オープンメッセージに対する応答メッセージに基づいて前記デバイスとの通信が可能な状態となる。

[0010] また、前記アプリケーションは、前記デバイスとの通信が可能な状態となった後、前記デバイスとの通信をクローズするクローズメッセージを含む前記リクエストを前記デバイス制御装置に送信する。

また、前記デバイス制御装置は、前記クローズメッセージを含む前記リクエストを受信し、前記クローズメッセージに対応する応答メッセージを含む前記レスポンスを前記アプリケーションに送信する。

[0011] また、前記アプリケーションは、前記デバイス制御装置で送信した前記クローズメッセージに対する応答メッセージを含む前記レスポンスを受信し、前記クローズメッセージに対する応答メッセージに基づいて前記デバイスとの通信をクローズする。

[0012] また、本発明のデバイス制御システムの制御方法は、データの入力もしくは出力を行うデバイス、アプリケーションを実行する端末、及び前記デバイスと接続するデバイス制御装置を有し、前記アプリケーションで生成したマークアップ言語のリクエストを前記デバイス制御装置に送信し、前記デバイス制御装置で前記リクエストを受信し、前記マークアップ言語のレスポンスを生成し、生成した前記レスポンスを前記端末へ送信することを特徴とする。

[0013] また、前記デバイス制御装置で、前記リクエストに含まれる情報を前記デバイスに送信し、前記デバイスは、前記リクエストに含まれる情報に基づいて処理を行い、前記リクエストに含まれる情報に基づいて処理した結果を前記デバイス制御装置に送信し、前記デバイスで送信した前記リクエストに含まれる情報に基づいて処理した結果を含む前記レスポンスを前記端末に送信する。

[0014] また、前記アプリケーションで、前記デバイス制御装置に通信経路の確立を要求する通信確立メッセージを含む前記リクエストを前記デバイス制御装置に送信し、前記デバイス制御装置で、前記通信確立メッセージを含む前記リクエストを受信した時、前記リクエストに含まれる前記通信確立メッセージに対する応答メッセージを含む前記レスポンスを前記アプリケーションに

送信する。

また、前記アプリケーションで、前記デバイス制御装置で送信された前記レスポンスに含まれる前記通信確立メッセージに対する応答メッセージに基づいて前記デバイス制御装置との前記通信経路を確立する。

[0015] また、前記通信経路が確立した後、前記アプリケーションで前記デバイス制御装置に前記デバイス制御装置に接続された前記デバイスとの通信をオープンするオープンメッセージを含む前記リクエストを前記デバイス制御装置に送信する。

また、前記デバイス制御装置で前記オープンメッセージを含む前記リクエストを受信した時、前記リクエストに含まれる前記オープンメッセージに対する応答メッセージを含む前記レスポンスを前記アプリケーションに送信する。

また、前記アプリケーションで前記デバイス制御装置が送信した前記オープンメッセージに対する応答メッセージを含む前記レスポンスを受信し、前記オープンメッセージに対する応答メッセージに基づいて前記デバイスとの通信が可能となる。

また、前記アプリケーションと前記デバイスとの通信が可能になった後に前記アプリケーションを停止させるとき、前記アプリケーションで前記デバイスとの通信をクローズするクローズメッセージを含む前記リクエストを前記デバイス制御装置に送信する。

また、前記デバイス制御装置で前記クローズメッセージを含む前記リクエストを受信し、前記クローズメッセージに対応する応答メッセージを含む前記レスポンスを前記アプリケーションに送信する。

また、前記アプリケーションで前記デバイス制御装置が送信した前記クローズメッセージに対する応答メッセージを含む前記レスポンスを受信し、前記クローズメッセージに対する応答メッセージに基づいて前記デバイスとの通信をクローズする。

図面の簡単な説明

- [0016] [図1]本発明の実施形態に係るデバイス制御システムの構成図である。
- [図2]デバイス制御システムの機能的構成を示す図である。
- [図3A]キーボードから取得可能なキーコードの一覧を示す図である。
- [図3B]キーボードから取得可能なキーコードの一覧を示す図である。
- [図4]インテリジェントプリンターに接続されるデバイスの接続例を示す図である。
- [図5]インテリジェントプリンターの外観斜視図である。
- [図6]コネクターパネルを示す図である。
- [図7]無線LANユニットの接続例を示す図である。
- [図8]環境構築のワークフローを示す図である。
- [図9]デバイス制御スクリプトの登録の手順を示す図である。
- [図10]制御されるデバイスの登録の手順を示す図である。
- [図11]登録されるデバイスの種類の一覧を示す図である。
- [図12]デバイスの登録の説明図である。
- [図13]デバイスの登録の説明図である。
- [図14]デバイスの登録の説明図である。
- [図15]デバイスの登録の説明図である。
- [図16]デバイスの登録の説明図である。
- [図17]基本的なプログラミングのシーケンスを示す図である。
- [図18]使用されるインターフェイスを示す図である。
- [図19A]データのフォーマットを示す図である。
- [図19B]データの内容を示す図である。
- [図20A]通信データの例を示す図である。
- [図20B]通信データの例を示す図である。
- [図21]メッセージの要素の配列の例を示す図である。
- [図22]メッセージを示す図である。
- [図23]デバイスごとのメッセージデータを示す図である。
- [図24A]<admin_info>メッセージの説明図である。

[図24B]<admin_info>メッセージの説明図である。

[図24C]<admin_info>メッセージの説明図である。

[図24D]<admin_info>メッセージの説明図である。

[図24E]<admin_info>メッセージの説明図である。

[図25A]<open_device>メッセージの説明図である。

[図25B]<open_device>メッセージの説明図である。

[図25C]<open_device>メッセージの説明図である。

[図25D]<open_device>メッセージの説明図である。

[図25E]<open_device>メッセージの説明図である。

[図25F]<open_device>メッセージの説明図である。

[図26A]<close_device>メッセージの説明図である。

[図26B]<close_device>メッセージの説明図である。

[図26C]<close_device>メッセージの説明図である。

[図26D]<close_device>メッセージの説明図である。

[図26E]<close_device>メッセージの説明図である。

[図27A]<device_data>メッセージの説明図である。

[図27B]<device_data>メッセージの説明図である。

[図27C]<device_data>メッセージの説明図である。

[図27D]<device_data>メッセージの説明図である。

[図28A]<error>メッセージの説明図である。

[図28B]<error>メッセージの説明図である。

[図28C]<error>メッセージの説明図である。

[図29A]バーコードスキャナー用メッセージの説明図である。

[図29B]バーコードスキャナー用メッセージの説明図である。

[図29C]バーコードスキャナー用メッセージの説明図である。

[図30A]プリンター用メッセージの説明図である。

[図30B]プリンター用メッセージの説明図である。

[図31A]プリンター用メッセージの説明図である。

[図31B]プリンター用メッセージの説明図である。

[図32A]プリンター制御XMLドキュメントの説明図である。

[図32B]プリンター制御XMLドキュメントの説明図である。

[図33A]プリンター制御XMLドキュメントの説明図である。

[図33B]プリンター制御XMLドキュメントの説明図である。

[図33C]プリンター制御XMLドキュメントの説明図である。

[図33D]プリンター制御XMLドキュメントの説明図である。

[図34A]プリンター制御XMLドキュメントの説明図である。

[図34B]プリンター制御XMLドキュメントの説明図である。

[図34C]プリンター制御XMLドキュメントの説明図である。

[図34D]プリンター制御XMLドキュメントの説明図である。

[図34E]プリンター制御XMLドキュメントの説明図である。

[図34F]プリンター制御XMLドキュメントの説明図である。

[図34G]プリンター制御XMLドキュメントの説明図である。

[図34H]プリンター制御XMLドキュメントの説明図である。

[図34I]プリンター制御XMLドキュメントの説明図である。

[図34J]プリンター制御XMLドキュメントの説明図である。

[図34K]プリンター制御XMLドキュメントの説明図である。

[図34L]プリンター制御XMLドキュメントの説明図である。

[図34M]プリンター制御XMLドキュメントの説明図である。

[図34N]プリンター制御XMLドキュメントの説明図である。

[図34P]プリンター制御XMLドキュメントの説明図である。

[図35]プリンター制御XMLドキュメントの説明図である。

[図36A]プリンター制御XMLドキュメントの説明図である。

[図36B]プリンター制御XMLドキュメントの説明図である。

[図36C]プリンター制御XMLドキュメントの説明図である。

[図36D]プリンター制御XMLドキュメントの説明図である。

[図37A]プリンター制御XMLドキュメントの説明図である。

[図37B]プリンター制御XMLドキュメントの説明図である。

[図38A]プリンター制御XMLドキュメントの説明図である。

[図38B]プリンター制御XMLドキュメントの説明図である。

[図38C]プリンター制御XMLドキュメントの説明図である。

[図38D]プリンター制御XMLドキュメントの説明図である。

[図38E]プリンター制御XMLドキュメントの説明図である。

[図38F]プリンター制御XMLドキュメントの説明図である。

[図38G]プリンター制御XMLドキュメントの説明図である。

[図38H]プリンター制御XMLドキュメントの説明図である。

[図38I]プリンター制御XMLドキュメントの説明図である。

[図38J]プリンター制御XMLドキュメントの説明図である。

[図39A]プリンター制御XMLドキュメントの説明図である。

[図39B]プリンター制御XMLドキュメントの説明図である。

[図39C]プリンター制御XMLドキュメントの説明図である。

[図39D]プリンター制御XMLドキュメントの説明図である。

[図39E]プリンター制御XMLドキュメントの説明図である。

[図39F]プリンター制御XMLドキュメントの説明図である。

[図39G]プリンター制御XMLドキュメントの説明図である。

[図39H]プリンター制御XMLドキュメントの説明図である。

[図39I]プリンター制御XMLドキュメントの説明図である。

[図39J]プリンター制御XMLドキュメントの説明図である。

[図39K]プリンター制御XMLドキュメントの説明図である。

[図40A]プリンター制御XMLドキュメントの説明図である。

[図40B]プリンター制御XMLドキュメントの説明図である。

[図41A]プリンター制御XMLドキュメントの説明図である。

[図41B]プリンター制御XMLドキュメントの説明図である。

[図42A]プリンター制御XMLドキュメントの説明図である。

[図42B]プリンター制御XMLドキュメントの説明図である。

- [図43A]プリンター制御XMLドキュメントの説明図である。
- [図43B]プリンター制御XMLドキュメントの説明図である。
- [図44A]プリンター制御XMLドキュメントの説明図である。
- [図44B]プリンター制御XMLドキュメントの説明図である。
- [図45A]プリンター制御XMLドキュメントの説明図である。
- [図45B]プリンター制御XMLドキュメントの説明図である。
- [図46]プリンター制御XMLドキュメントの説明図である。
- [図47A]プリンター制御XMLドキュメントの説明図である。
- [図47B]プリンター制御XMLドキュメントの説明図である。
- [図48A]プリンター制御XMLドキュメントの説明図である。
- [図48B]プリンター制御XMLドキュメントの説明図である。
- [図49A]プリンター制御XMLドキュメントの説明図である。
- [図49B]プリンター制御XMLドキュメントの説明図である。
- [図50A]プリンター制御XMLドキュメントの説明図である。
- [図50B]プリンター制御XMLドキュメントの説明図である。
- [図50C]プリンター制御XMLドキュメントの説明図である。
- [図51A]プリンター制御XMLドキュメントの説明図である。
- [図51B]プリンター制御XMLドキュメントの説明図である。
- [図51C]プリンター制御XMLドキュメントの説明図である。
- [図52A]カスタマーディスプレイ用メッセージの説明図である。
- [図52B]カスタマーディスプレイ用メッセージの説明図である。
- [図53A]カスタマーディスプレイ用メッセージの説明図である。
- [図53B]カスタマーディスプレイ用メッセージの説明図である。
- [図54A]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図54B]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図55A]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図55B]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図55C]カスタマーディスプレイ制御XMLドキュメントの説明図である。

- [図56]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図57A]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図57B]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図57C]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図57D]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図58A]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図58B]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図58C]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図59A]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図59B]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図59C]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図60A]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図60B]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図61A]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図61B]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図62A]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図62B]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図62C]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図62D]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図62E]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図62F]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図63]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図64]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図65]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図66]カスタマーディスプレイ制御XMLドキュメントの説明図である。
- [図67A]POSキーボード用メッセージの説明図である。
- [図67B]POSキーボード用メッセージの説明図である。
- [図67C]POSキーボード用メッセージの説明図である。

- [図68A] P O S キーボード用メッセージの説明図である。
- [図68B] P O S キーボード用メッセージの説明図である。
- [図69A] P O S キーボード用メッセージの説明図である。
- [図69B] P O S キーボード用メッセージの説明図である。
- [図69C] P O S キーボード用メッセージの説明図である。
- [図70A] シリアル通信デバイス用メッセージの説明図である。
- [図70B] シリアル通信デバイス用メッセージの説明図である。
- [図71A] シリアル通信デバイス用メッセージの説明図である。
- [図71B] シリアル通信デバイス用メッセージの説明図である。
- [図71C] シリアル通信デバイス用メッセージの説明図である。
- [図72] デバイス制御スクリプトの説明図である。
- [図73] デバイス制御スクリプトの説明図である。
- [図74] デバイス制御スクリプトの説明図である。
- [図75] デバイス制御スクリプトの説明図である。
- [図76] デバイス制御スクリプトの説明図である。
- [図77] デバイス制御スクリプトの説明図である。
- [図78] デバイス制御スクリプトの説明図である。
- [図79] デバイス制御スクリプトの説明図である。
- [図80] デバイス制御スクリプトの説明図である。
- [図81] デバイス制御スクリプトの説明図である。
- [図82] プログラムの例を示す図である。
- [図83] プログラムの環境構築のフローを示す。
- [図84A] プログラムの環境構築の説明図である。
- [図84B] プログラムの環境構築の説明図である。

発明を実施するための形態

[0017] 以下、図面を参照して本発明の実施形態について説明する。

図1は、本発明を適用した実施形態のデバイス制御システム100の構成を示す図である。デバイス制御システム100は、端末3、及び、プリンタ

ー（インテリジェントプリンター）５を備えている。

端末３は、ネットワークに接続して、Socket通信ができるデバイスであればよく、パーソナルコンピューター、スマートフォン、あるいはタブレットコンピューターなどが使用できる。端末３は、Socket通信ができ、マークアップ言語であるXML文書をハンドリングできる環境を有し、OS（オペレーティングシステム）として、例えば、iOS（登録商標）、Android（登録商標）、Windows（登録商標）、Linux（登録商標）、Mac（登録商標）OS等を備える。

[0018] 端末３及びプリンター５は通信ネットワークを介して接続される。図１の例では、デバイス制御システム１００は、無線LANアクセスポイント１１を有し、プリンター５に無線LANアクセスポイント１１がケーブル接続される。端末３と無線LANアクセスポイント１１とが無線接続されることにより、端末３とプリンター５が通信する。

[0019] プリンター５は、制御対象のデバイスとして、ネットワークプリンター７、カスタマーディスプレイ８、及びバーコードスキャナー９が接続される。ネットワークプリンター７は、プリンター５にネットワークを介して接続される。ネットワークプリンター７は、プリンター５が備えるローカルプリンター５５（図２）とは別のプリンターであり、有線LANまたは無線LANのインターフェイスを備える。カスタマーディスプレイ８は、後述するUSBインターフェイスに接続される。バーコードスキャナー９は、後述するUSBインターフェイスを介して接続される。これらのデバイスは、一般的に周辺機器（peripheral）と呼ばれるが、以下の説明ではデバイスと呼ぶ。プリンター５に接続できるデバイスは図１に示す機器に限定されない。例えば、ディスプレイ、及び、キーボード等のキー入力デバイスを含む。さらに、OS標準のHID（Human interface device）ドライバーで制御可能なデバイス、OS標準のシリアル通信ドライバーで動作可能なシリアル通信デバイス、及び、シリアル通信デバイスと同様に制御可能なUSBデバイスを含む。

[0020] デバイス制御システム１００の動作は、次の通りである。

- (1) 端末3にアプリケーションを配置し、表示する。
- (2) アプリケーションからプリンター5にリクエストメッセージを送信する。
- (3) プリンター5から制御可能なデバイスにデータを送信する。
- (4) プリンター5に接続されているデバイスやネットワークプリンターを制御する。
- (5) プリンター5が端末3にレスポンスを返す。

[0021] 図2は、デバイス制御システム100の機能的構成を示す図である。

デバイス制御システム100は、プリンター5に接続されたデバイス（周辺機器）を制御する機能をマークアップ言語であるXMLのコマンド体系であるデバイスXMLを備える。

端末3は、アプリケーション31を有する。アプリケーション31は、端末3にインストールされたネイティブアプリケーションプログラムである。アプリケーション31は、デバイスアプリケーションプログラムインターフェース（API）33は、プリンター5に対して、XMLフォーマットのリクエストメッセージをSocket通信で送信する（Request）。

[0022] プリンター5には、デバイスサービスインターフェイス501及びデバイス制御スクリプト502が組み込まれる。デバイスサービスインターフェイス501及びデバイス制御スクリプト502は、リクエストメッセージを解釈して、デバイス制御スクリプト502にリクエストを出力する。デバイス制御スクリプト502は、キー入力デバイス18及びシリアル通信デバイス19を制御する。デバイス制御スクリプト502は、キー入力デバイス18及びシリアル通信デバイス19のイベントや制御結果のレスポンスを、デバイスサービスインターフェイス501に出力する。ここで、キー入力デバイス18は図1のキーボード14を含み、シリアル通信デバイス19は図1のバーコードスキャナー9及びキャッシュドローワー13を含む。

また、デバイスサービスインターフェイス501は、ローカルプリンター55とデータを送受信する。ローカルプリンター55は、プリンター5が備

えるプリンターである。また、デバイスサービスインターフェイス501は、ネットワークプリンター7及びカスタマーディスプレイ8とデータを送受信する。

デバイスサービスインターフェイス501は、ネットワークプリンター7、カスタマーディスプレイ8及びローカルプリンター55の制御の結果やデバイス制御スクリプト502のレスポンスをもとに、XMLフォーマットのレスポンス(Response)をアプリケーション31に出力する。

このように、プリンター5は、デバイスサービスインターフェイス501及びデバイス制御スクリプト502によってデバイスを制御する。図2の例で制御されるデバイスはネットワークプリンター7、カスタマーディスプレイ8、キー入力デバイス18、シリアル通信デバイス19、及び、ローカルプリンター55である。

[0023] 以上の動作により、デバイス制御システム100が備えるデバイスXMLを利用して、端末3によってプリンター5に接続されたデバイスを制御できる。

デバイスXMLは、以下の特徴を有する。

- ・デバイスの制御用に別途、コントローラーとなる機器（コンピューターなど）を用意する必要がない。

- ・プリンター5のOSが備える標準のドライバーで動作するデバイスは、ドライバーをインストールせずに、デバイス制御スクリプト502で利用できる。

- ・デバイスにアクセスすると、デバイスは、排他ロックされる。複数の端末3から同時アクセスしても、制御が混在しない。デバイスを制御していた端末3がデバイスを解放すると、解放されたデバイスは他の端末3から制御可能となる。

[0024] また、デバイスAPI33は、以下の特徴を有する。

- ・罫線を印刷するコマンドは、プリンター5のローカルプリンター55またはネットワークプリンター7が「罫線コマンド」をサポートしている場合

に使用できる。

- ・プリンター 5 またはネットワークプリンター 7 にブザーが接続されている場合にブザーの機能を使用できる。

- ・キーボード 1 4 から取得できるキーコードに制限がある。キーボード 1 4 から取得可能なキーコードは、図 3 A、図 3 B の通りである。

[0025] プリンター 5 の構成について説明する。

プリンター 5 は、CPU、RAM、フラッシュROM、NV（不揮発性メモリー）、ビデオコントローラー、補助記憶装置（SSD：solidstatedrive）、インターフェイス、及び、ローカルプリンター（印刷部）を備えている。プリンター 5 は、スピーカーを備えていてもよい。ローカルプリンターは、80mm幅または58mm幅のロール紙に印刷するサーマルプリンターである。

プリンター 5 にはPOS（point of sales）システムをインストールできる。プリンター 5 のOS（operating system）は、例えばWindows（登録商標）ベースであり、補助記憶装置に記憶される。

[0026] プリンター 5 は、プリンター 5 に接続されるデバイスを端末 3 が制御するソフトウェアである、デバイス制御プログラムを備えている。このため、端末 3 はドライバプログラムを必要としない。

プリンター 5 は、プリンター 5 が備えるローカルプリンター、及び、デバイスを制御するソフトウェアである、Windows（登録商標）標準のデバイスドライバプログラム（APD）、UPOSドライバ、OPOSドライバ等を備えている。

プリンター 5 には、ウェブアプリケーションをインストールできる。これにより、図 3 に示すようにプリンター 5 をアプリケーションサーバーとして利用できる。ウェブアプリケーションは、例えば、PHP及びPerlのサーバーサイドスクリプト、SQLiteデータベースアクセス（サーバーサイドスクリプト）が可能である。

[0027] 図 4 は、プリンター 5 に接続されるデバイスの接続例を示す図である。

プリンター 5 には、上述したネットワークプリンター 7、カスタマーディ

スプレイ 8 及びバーコードスキャナー 9 を接続できる。プリンター 5 にディスプレイ 12、キャッシュドローワー 13、及びキーボード 14 を接続できる。

図 5 は、プリンター 5 の外観斜視図である。

プリンター 5 の上部には、ロール紙カバー 51 が設けられる。ロール紙カバーはカバーオープンボタン 52 を押すと開き、ロール紙を出し入れできる。印刷後のロール紙を排出する排出口には、手動でロール紙をカットするマニュアルカッター 53、及び、カッターカバー 54 を有する。カッターカバー 54 はプリンター 5 のローカルプリンターで紙ジャムが発生した場合及びロール紙カバー 51 が開かない場合に開かれる。カッターカバー 54 を開くとマニュアルカッター 53 のブレードが基準位置に戻る。プリンター 5 は、電源スイッチ 56B、リセットボタン 56A、LED 表示部 57、及び、コントロールパネル 58 を備える。LED 表示部 57 は、補助記憶装置のアクセスを示すディスクアクセス LED、及び、ステータス LED を備える。ステータス LED は、OS の動作状態、OS のスタンバイモード、電源オフ状態、OS の起動シーケンス中、CPU の高温警告を通知する。

コントロールパネル 58 はパワー LED、エラー LED、ロール紙 LED、及び送りボタンを備える。パワー LED は電源が供給されている場合に点灯する。エラー LED は通常動作時は消灯し、リセットされたとき、及び、ロール紙の末端が検出されて印刷が停止したときに点灯する。ペーパー LED はロール紙の残量が十分なときは消灯し、残量がすくなくなると点灯し、自己診断機能の実行時に点滅する。送りボタンを押すと 1 行ずつ、または連続してロール紙が搬送される。

[0028] プリンター 5 の後部にはコネクタカバー 59 が設けられる。コネクタカバー 59 を外すと、プリンター 5 の背面のコネクターパネル (connection panel) が現れる。

図 6 はコネクターパネル 60 を示す図である。コネクターパネル 60 は、ドローワーキックコネクター 61、イーサネットコネクター 62、USB コネ

クター63、VGAコネクター64、COMコネクター65、ライン出力66、及びDC入力67を有する。コネクターパネル60は、本発明の接続部に相当する。但し、プリンター5のデバイスには、プリンター5本体が備えるローカルプリンター55が含まれる。従って、接続部は、コネクターパネル60、及び、プリンター5内部のインターフェイスを含む。

ドロワーキックコネクター61には、キャッシュドロワー13またはオプションのブザーが接続される。イーサネットコネクター62はネットワークに接続される。USBコネクター63は6個のUSBポートを有する。USBコネクター63にはカスタマーディスプレイ8、バーコードスキャナー9、キーボード14及びその他のデバイスが接続される。VGAコネクター64にはディスプレイ12が接続される。COMコネクター65にはシリアル通信を行うデバイスのシリアルインターフェイスが接続される。ライン出力66は外部のスピーカーに接続される。

図7は、無線LANユニット17の接続例を示す図である。無線LANユニット17は、USB延長ケーブル15に差し込まれ、USB延長ケーブル15はUSBコネクター63に差し込まれる。これにより、プリンター5を無線LAN接続することができる。

[0029] デバイスサービスインターフェイス501の環境構築について説明する。

図8は、環境構築のワークフローを示す。

1. ネットワーク設定

システム及びプリンター5のネットワーク設定をする。プリンター5のネットワーク設定は、以下のいずれかの方法により設定できる。

- ・プリンター5で、Windows（登録商標）のネットワーク設定を行う。
- ・外部機器（Windowsのコンピューター）に、プリンター5の設定アプリケーションであるTMNet WinConfigをインストールして設定する。

2. プリンター5にデバイスを接続

プリンター5にデバイスを接続する。接続可能なデバイスは、上述したように、カスタマーディスプレイ8、バーコードスキャナー9、ディスプレイ

12、キャッシュドローワー13、キーボード14等である。さらに、OS標準のHIDドライバで制御可能なデバイス、OS標準のシリアル通信ドライバで動作可能なシリアル通信デバイス、及び、シリアル通信デバイスと同様に制御可能なUSBデバイスを含む。図6のコネクターパネル60には、COMコネクター65が1つしかないが、シリアルーUSB変換ケーブル、及び、シリアルーUSB変換に対応したドライバプログラムがあれば、複数のシリアル通信デバイスを接続できる。

3. デバイス制御スクリプトの登録

プリンター5が対応した製品以外のデバイスをプリンター5で制御する場合、ユーザーが用意したデバイス制御スクリプトを登録する。登録は、ウェブブラウザから行う。

4. デバイスの登録

プリンター5に接続するデバイスをプリンター5のソフトウェアに登録する。登録は、ウェブブラウザから行う。

手順1. ～5で使用するウェブブラウザは、端末3で動作するウェブブラウザを含む。

[0030] 図9を参照して、ユーザーが予め用意したデバイス制御スクリプトを登録する方法を説明する。

登録の手順は以下の通りである。

(1) カスタマイズしたデバイス制御スクリプトを用意する。

(2) プリンター5の電源をオンにする。

(3) ウェブブラウザを起動し、URL (<http://プリンター5のIPアドレス/WebConfig/>) を入力する。

(4) TMNetWebConfigが起動する。図9の画面の[Webサービス設定] - [登録と削除] をクリックする。

(5) “制御スクリプト” 画面が表示される。

[登録する制御スクリプト] の[参照] をクリックし、登録するデバイス制御スクリプトを選択する。

(6) [登録] をクリックする。

デバイス制御スクリプトが登録され、画面下の [登録済み制御スクリプト] に一覧表示される。

[0031] デバイスXMLにより制御されるデバイスを登録する。図10に示すように、ウェブブラウザでTMNetWebConfigを表示してデバイスを登録する。

登録の手順は以下の通りである。

(1) ウェブブラウザを起動し、URL (http://プリンター5のIPアドレス/WebConfig/) を入力する。

(2) TMNetWebConfigが起動する。図10の画面の [Webサービス設定] - [デバイス登録] より登録するデバイスの種類にしたがってクリックする。図10で [デバイス登録] にはデバイスが表示される。登録するデバイスの種類をクリックする。本実施形態ではデバイスの種類は、図11に示すように、プリンター、ディスプレイ、キー入力デバイス、シリアル通信デバイス、その他、の5種類から選択される。デバイスの種類の詳細は図11に示す通りである。

[0032] (3) 各デバイスの設定画面が表示される。接続したデバイスをプリンター5に登録する。

(3-1) プリンターを登録する場合

プリンター5の本体が備えるローカルプリンターが登録されていることを、以下の手順で確認する。

(3-1-1) 図10の画面で [登録済みプリンター] のデバイスID "local_printer" に、"プリンター5" が登録されていることを確認する。

(3-1-2) [テスト印字] をクリックする。ローカルプリンター55により「TEST_PRINT」が印字されることを確認する。

(3-1-3) 接続しているネットワークプリンター7を登録する。

以下の手順で登録する。

(3-1-3-1) 図12に示す項目 (デバイスID、タイプ、型番、IPアドレス、リトライ間隔) を設定し、[登録] をクリックする。

(3-1-3-2) [登録済みプリンター] に追加されたことを確認したら、[テスト印字] をクリックする。登録したネットワークプリンター 7 により「TEST_PRINT」が印字されることを確認する。

(3-2) ディスプレイ (カスタマーディスプレイ 8) を登録する場合。
カスタマーディスプレイ 8 のデバイス ID は、"local_display" 固定である。
接続しているカスタマーディスプレイ 8 を、以下の手順で登録する。

(3-2-1) 図 10 の画面で [使用する] を選択する。

(3-2-2) 図 13 の項目 (通信設定、輝度設定) を設定し、[登録] をクリックする。

(3-2-3) [テスト表示] をクリックする。カスタマーディスプレイ 8 に文字が表示されることを確認する。

(3-3) キー入力デバイス 18 を登録する場合

接続しているキー入力デバイス 18 を、以下の手順で登録する。

(3-3-1) 図 14 の項目 (デバイス ID、デバイス名、制御スクリプト) を設定し、[登録] をクリックする。

(3-3-2) [登録済みキー入力デバイス] に追加されたことを確認したら、[動作テスト] をクリックする。

(3-3-3) ウェブブラウザにより画面が表示される。キー入力デバイス 18 を操作し、操作したとおりの結果が表示されることを確認する。

(3-4) シリアル通信デバイス 19 を登録する場合

接続しているシリアル通信デバイス 19 を、以下の手順で登録する。

(3-4-1) 接続しているシリアル通信デバイス 19 について、図 15 の項目 (デバイス ID、デバイス名、制御スクリプト、通信速度、データビット、パリティ、ストップビット、フリー制御) を設定し、[登録] をクリックする。

(3-4-2) [登録済みシリアル通信デバイス] に追加されたことを確認する。

(3-5) その他のデバイスを登録する場合

接続しているデバイスを、以下の手順で登録する。

(3-5-1) 接続しているデバイスについて、図16の項目(デバイスID、制御スクリプト)を設定し、[登録]をクリックする。

(3-5-2) [登録済みその他デバイス]に追加されたことを確認する。

[0033] ここで、Device XMLを使用したアプリケーション開発のプログラミング方法について説明する。

Device XMLの基本的なプログラミングのシーケンスは、図17のとおりである。

図17の”Applications”はアプリケーション31に相当し、”DeviceControl”はデバイス制御機能(デバイスサービスインターフェイス501)に相当し、”Device”は制御されるデバイスを示す。

[0034] 1: TCP接続の確立。アプリケーション31からプリンター5のSocket I/FにTCP接続する。

2: Connectメッセージ。デバイスサービスインターフェイス501との通信経路を確立する。

3: アプリケーション31がopen_deviceメッセージ(type_printer)を送信する。

3.1: デバイスサービスインターフェイス501によりデバイスに”Cmd ()”コマンドが送信される。

4: デバイスは”Deviceopen”を返す。

4.1: デバイスサービスインターフェイス501がレスポンスを返す。

6: アプリケーション31がdevice_data_messageを送信する。device_data_messageを使ってデバイスを制御する。

6.1: デバイスに”Cmd ()”コマンドが送信される。

6.1.1: デバイスが実行する。

ここで、Device_data_messageにレスポンスする。

7: デバイスが実行結果を返す。

7. 1 : デバイスサービスインターフェイス 5 0 1 が device_data_message、または、error message を返す。

8 : アプリケーション 3 1 が close_device_nmessage を送信する。デバイスサービスインターフェイス 5 0 1 との接続および Socket I/F 通信を切断する。

8. 1 : デバイスサービスインターフェイス 5 0 1 によりデバイスに " Cmd () " コマンドが送信される。

9 : デバイスが close する。

9. 1 : デバイスサービスインターフェイス 5 0 1 が アプリケーション 3 1 に レスポンス を返す。

[0035] デバイス XML による制御を行う場合、図 1 8 のインターフェイスが使用される。

各データアイテムのフォーマットは図 1 9 A の通りであり、各データの内容は図 1 9 B に示す。

[0036] アプリケーション 3 1 とデバイスサービスインターフェイス 5 0 1 との間の通信データの例を図 2 0 A、図 2 0 B に示す。図 2 0 A、図 2 0 B では、NULL 文字を " \ 0 " で表現する。図 2 0 A、図 2 0 B には、接続の確立、管理者情報の取得、デバイスのオープン、デバイスの制御、デバイスのクローズのデータの例を示す。

[0037] <device_data>メッセージの<data>要素の配列方法を説明する。

<data>タグ内で配列を記述する際のプログラミング方法は以下のとおりである。

- ・配列にする要素の先頭に属性 array="true" を指定する。
- ・以降は同名の要素名で値を指定する。

keycodes= [49, 50, 51, 52] ; を表現する場合の例を図 2 1 に示す。

[0038] Device XML について説明する。ePOS-Device XML の一覧を示す。

ePOS-Device XML には、以下が用意されている

- ・メッセージ (図 2 2)

メッセージは、アプリケーション 31 からプリンター 5 にリクエストするメッセージ、及び、プリンター 5 からアプリケーション 31 へのレスポンスメッセージのXMLである。

・デバイスごとのメッセージデータ（図 23）

制御するデバイスごとの、リクエスト、及びレスポンスのデータを格納するXMLである。この形式は、<device_data>メッセージの<data>要素の子要素になる。

[0039] デバイスごとのメッセージデータについて説明する。

デバイスごとのメッセージデータは、<device_data>メッセージの<data>要素の子要素で、デバイスを制御するデータを指定する。

メッセージデータの構成要素はデバイスの種類に応じて異なる。

データを指定する前に、<type>要素でメッセージデータの種類を指定してから、子要素のデータを指定する。デバイスの種類ごとに使用可能な<type>要素は図 23 の通りである。

[0040] 以下、メッセージについて説明する。

<connect>

[レスポンス] に含まれるメッセージである。

コネクションが確立されたことを、アプリケーションに返す。

アプリケーション 31 は、本メッセージの受信をトリガーとして、以降に説明するメッセージ送信を行う。

[例]

<connect></connect>

[0041] <admin_info>

プリンター 5 に設定された、管理者情報を返す。TMNet WebConfigで設定した情報である。

[リクエスト] に含まれるメッセージである。

管理者情報を、プリンター 5 へ要求する。

[例]

<admin_info></admin_info>

[レスポンス] に含まれるメッセージである。

管理者情報を、アプリケーション 31 に返す。子要素、データ型及び説明を図 2 4 A に示す。

- ・ code

code要素を図 2 4 B に示す。

- ・ data

data要素を図 2 4 C に示す。

受信データの文字列について述べる。図 2 4 D に示す文字列はエスケープ処理される。

[例] 正常時および異常時のレスポンスの例を、図 2 4 E に示す。

[0042] <open_device>

デバイスIDに紐付いたデバイスを使用可能にする。リクエストしたアプリケーション 31 に対し、指定されたデバイスの排他的使用权を与える。

[リクエスト] に含まれるメッセージである。

オープンするデバイスを指定する。子要素、データ型及び説明を図 2 5 A に示す。

- ・ data

data要素を図 2 5 B に示す。

[例] リクエストの例を図 2 5 C に示す。

[レスポンス] に含まれるメッセージである。

デバイスのオープン結果を、アプリケーション 31 に返す。子要素、データ型及び説明を図 2 5 D に示す。

- ・ code

code要素を図 2 5 E に示す。

[例] 正常時および異常時のレスポンスの例を、図 2 5 F に示す。

[0043] <close_device>

オープンしたデバイスをクローズする。

〔リクエスト〕に含まれるメッセージである。

クローズするデバイスを指定する。子要素、データ型及び説明を図 2 6 A に示す。

〔例〕 リクエストの例を図 2 6 B に示す。

〔レスポンス〕に含まれるメッセージである。

デバイスのクローズ結果を、アプリケーション 3 1 に返す。子要素、データ型及び説明を図 2 6 C に示す。

・ code

code要素を図 2 6 D に示す。

〔例〕 正常時および異常時のレスポンスの例を、図 2 6 E に示す。

[0044] <device_data>

〔リクエスト〕に含まれるメッセージである。

デバイスへのデータを送信する。デバイス制御の命令や印刷・表示データが含まれる。子要素、データ型及び説明を図 2 7 A に示す。

〔例〕 リクエストの例を図 2 7 B に示す。

〔レスポンス〕に含まれるメッセージである。

デバイスからのデータを返す。デバイス制御の結果、デバイスで発生したイベントや、デバイスからの入力データが含まれる。子要素、データ型及び説明を図 2 7 C に示す。

〔例〕 レスポンスの例を図 2 7 D に示す。

[0045] <error>

〔レスポンス〕に含まれるメッセージである。

デバイス制御命令、サービス制御命令でのエラー、その他共通エラーを返す。子要素、データ型及び説明を図 2 8 A に示す。

・ code

code要素を図 2 8 B に示す。

〔例〕 レスポンスの例を図 2 8 C に示す。

[0046] バーコードスキャナーのメッセージデータについて説明する。

<type>ondata</type>

[レスポンス] に含まれるメッセージである。

バーコードスキャナーからのスキャンデータを、アプリケーションへ返す。子要素、データ型及び説明を図 29 A に示す。

受信データの文字列について述べる。図 29 B の文字列はエスケープ処理される。

[例] レスポンスの例を図 29 C に示す。

[0047] プリンターのメッセージデータについて説明する。

印刷データについては、後述の「プリンター制御用XML」で説明する。

<type>print</type>

[リクエスト] に含まれるメッセージである。

プリンターへ印刷データと設定データを送信する。子要素、データ型及び説明を図 30 A に示す。

[例] リクエストの例を図 30 B に示す。

[0048] <type>onxmlresult</type>

[レスポンス] に含まれるメッセージである。

印刷結果を、アプリケーションへ返す。子要素、データ型及び説明を図 31 A に示す。

[例] レスポンスの例を図 31 B に示す。

[0049] プリンターを制御するXML文書であるプリンター制御用XMLについて説明する。

プリンターのメッセージデータについては、後述する「プリンターのメッセージデータ」で説明する。

<epos-print>

アプリケーションからプリンターへ送信するXML文書である。指定した機能の実行をプリンターにリクエストする。<epos-print>は要素を持っている。

[属性]

- ・ xmlns

epos-printの名前を宣言する。

[子要素] 子要素を図 3 2 A に示す。

[例] XML 文書の例を図 3 2 B に示す。

[0050] <response>

プリンターからアプリケーションへ返す XML 文書である。

[属性]

- ・ success

表示結果を取得する。属性値を図 3 3 A に示す。

- ・ code

エラーコードを取得する。属性値を図 3 2 B に示す。

- ・ status

プリンターの状態に応じて、以下の値の論理和を10進表現で取得する。属性値を図 3 2 C に示す。

[例] XML 文書の例を図 3 3 D に示す。

[0051] <text>

印字する文字列を指定する。また、文字の装飾や印字位置、改行量など文字列に関する設定もする。

テキストの印字後、テキスト以外を印字する場合、改行または紙送りを実行する。

ページモードでは、文字のベースラインドット（151ページ「プリンターの仕様」）を基準として、現在の印字位置に文字を展開する。

[実体参照]

プリンター制御に必要な水平タブ、改行、および以下の記号は、図 3 4 A の実体参照を用いて記述する。

[属性]

- ・ lang

対象言語を指定する。属性値を図 3 4 B に示す。

印字できる文字コードは、各プリンターの仕様による。

・ font

文字フォントを指定する。属性値を図 3 4 C に示す。

日本語を指定すると、シフトJISコード体系の文字を印字できる。

・ smooth

スムージングを設定する。スムージングを有効にすると、文字の印字品質が向上する。属性値を図 3 4 D に示す。

・ dw

横倍角を設定する。属性値を図 3 4 E に示す。

ひとつの要素の中にdw属性とwidth属性を同時に指定した場合、width属性の倍率指定を優先される。

・ dh

縦倍角を設定する。属性値を図 3 4 F に示す。

ひとつの要素の中にdh属性とheight属性を同時に指定した場合、height属性の倍率指定を優先される。

・ width

横方向倍率を指定する。属性値を図 3 4 G に示す。

ひとつの要素の中にdw属性とwidth属性を同時に指定した場合、width属性の倍率指定を優先される。

・ height

縦方向倍率を指定する。属性値を図 3 4 H に示す。

ひとつの要素の中にdh属性とheight属性を同時に指定した場合、height属性の倍率指定を優先される。

・ reverse

白黒反転文字を設定する。属性値を図 3 4 I に示す。

・ ul

アンダーラインを設定する。属性値を図 3 4 J に示す。

・ em

強調印字を設定する。属性値を図 3 4 K に示す。

・ color

文字色を指定する。属性値を図 3 4 L に示す。

・ x

(初期値 : “0”)

文字の印字開始位置をドット単位で指定する。

ページモードでは、position要素による横方向の印字位置指定と同じであるが、横方向の印字位置が移動する。

・ align

位置揃えを指定する。属性値を図 3 4 M に示す。

スタンダードモードの場合、align属性の指定は「行の先頭の状態」で行う。

本要素で設定するalign属性は、image, logo, barcode, symbol要素のalign属性にも適用される。

・ rotate

倒立印字を設定する。属性値を図 3 4 N に示す。

ページモードでは、rotateの指定は無効である。

ページモードではdirectionで印字方向を右→左 (right_to_left) に設定して180度回転印字を行う。

本要素で設定するrotate属性は、barcode, symbol要素のrotate属性にも適用される。

・ linespc

(初期値 : “30”)

1行あたりの紙送り量をドット単位で指定する。

[例] 文字列を印字する設定の例を図 3 4 P に示す。

[0052] <feed>

紙送り量を指定する。ドット単位または、行単位で紙送り量を指定する。紙送り量の指定がない場合、1行の紙送り（改行）する。また、1行あたりの

改行量を設定する。

[属性]

・ unit

紙送り量をドット単位で指定する。

・ line

紙送り量を行単位で指定する。

・ linespc

(初期値: “30”)

1行あたりの紙送り量をドット単位で指定する。

1行あたりの紙送り量が30ドットの場合、最大240行まで指定可能である。

スタンダードモードとページモードでは、改行量を別々に保持する。linespc属性を指定すると、後続く<text>、<feed>のlinespc属性に影響する。

[例] XML文書の例を図35に示す。

[0053] <image>

ラスター形式のビットイメージデータを指定する。(データ型xs:base64Binary)

ラスター形式とは、画像の左上を原点に画素を水平方向に走査したデータである。

データは2階調の場合は1画素あたり1ビット、16階調の場合は1画素あたり4ビットで、バイトデータの上位ビットから詰めていく。また、1ラインの走査データがバイト単位になるように、0でパディングする。

ラスターイメージを高速に印字する場合、align属性をleftに指定し、width属性の値をプリンターの用紙幅を超えない8の倍数に指定する。

ページモードでは、ラスターイメージの左下ドットを基準として、現在の印字位置にラスターイメージを展開する。印字位置は移動しない。

ページモードでは、ラスターイメージが印字領域をはみ出さないように、印字位置を設定する。

多階調ラスターイメージで印刷したバーコード・2次元コードの読み取り品

質の影響があるので、2階調で印刷する。

ラスター形式のビットイメージは、ePOS-Print XML生成ツールで作成するか、ユーザーのアプリケーションで作成する。ユーザーのアプリケーションで作成する場合次の点を考慮する。

- ・ 2階調の場合：画像の幅を8の倍数にするか、足りないbitを0にする。
- ・ 16階調の場合：画像の幅を2の倍数にするか、足りないbitを0にする。

[属性]

・ width

画像の幅をドット単位で指定する。

・ height

画像の高さをドット単位で指定する。

・ color

文字色を指定する。属性値を図 3 6 A に示す。

・ align

印刷位置を指定する。属性値を図 3 6 B に示す。

本要素で設定するalign属性は、text, logo, barcode, symbol要素のalign属性にも適用される。

・ mode

(オプション (オプション))

カラーモードを指定する。属性値を図 3 6 C に示す。

[例] XML 文書の例を図 3 6 D に示す。

[0054] <logo>

プリンターのNVメモリーに登録されているロゴを指定する。

ロゴは、機種専用ユーティリティーやロゴ登録ユーティリティー (TMFLogo) を使って、あらかじめプリンターにロゴの登録をする。

ページモードでは、ロゴの左下ドットを基準として、現在の印字位置にロゴを展開する。

[属性]

- ・ key1

NVロゴ登録時に設定したキーコード1の値を指定する。

- ・ key2

NVロゴ登録時に設定したキーコード2の値を指定する。必ず指定する。

- ・ align

印刷位置を指定する。属性値を図 3 7 A に示す。

本要素で設定するalign属性は、text, image, barcode, symbol要素のalign属性にも適用される。

[例] XML 文書の例を図 3 7 B に示す。

[0055] <barcode>

バーコードデータを文字列で指定する。

バーコードの規格に準拠しない設定を行った場合、あるいは、バーコードがプリンターの印字領域よりも大きくなる場合、バーコードを印字しない。

ページモードでは、バーコードの左下ドット（HRIを除く）を基準として、現在の印字位置にバーコードを展開する。

[バーコードの種類]

図 3 8 A、図 3 8 B、図 3 8 C に示すバーコードを指定することができる。

文字列で表現できないバイナリーデータを指定する場合、図 3 8 D に示すエスケープシーケンスで指定する。

[属性]

- ・ type

バーコードの種類を指定する。属性値を図 3 8 E に示す。

- ・ hri

HRIの位置を指定する。属性値を図 3 8 F に示す。

- ・ font

HRIフォントを指定する。属性値を図 3 8 G に示す。

- ・ width

(初期値：“3”)

モジュールの幅をドット単位で指定する。2～6の整数値で指定する。

・ height

(初期値：“162”)

モジュールの高さをドット単位で指定する。

・ align

印刷位置を指定する。属性値を図 3 8 H に示す。

本要素で設定するalign属性は、text, image, logo, symbol要素のalign属性にも適用される。

・ rotate

倒立印字を指定する。属性値を図 3 8 I に示す。

本要素で設定するrotate属性は、text, symbol要素のrotate属性にも適用される。

[例] XML 文書の例を図 3 8 J に示す。

[0056] <symbol>

2次元コードを印刷する。

2次元コードを文字列で指定する。

2次元コードの規格に準拠しない設定を行った場合、あるいは、2次元コードがプリンターの印字領域よりも大きくなる場合、2次元コードを印字しない。

ページモードでは、2次元コードの左上ドットを基準として、現在の印字位置に2次元コードを展開する。

[2次元コードの種類]

図 3 9 A、図 3 9 B に示す2次元コードを指定することができる。

文字列で表現できないバイナリーデータを指定する場合、図 3 9 C に示すエスケープシーケンスで指定する。

[属性]

・ type

2次元コードの種類を指定する。属性値を図39Dに示す。

・ level

エラー訂正レベルを指定する。属性値を図39Eに示す。

レベルは2次元コードの種類に合わせて選択する。

MaxiCode, 2次元GS1DataBarの場合、defaultを選択する。

・ width

1モジュールの幅をドット単位で指定する。詳細を図39Fに示す。

・ height

1モジュールの高さをドット単位で指定する。詳細を図39Gに示す。

・ size

図39Hに詳細を示す。

・ align

印刷位置を指定する。属性値を図39Iに示す。

本要素で設定するalign属性は、text, image, logo, barcode要素のalign属性にも適用される。

・ rotate

倒立印字を指定する。属性値を図39Jに示す。

本要素で設定するrotate属性は、text, barcode要素のrotate属性にも適用される。

[例] XML文書の例を図39Kに示す。

[0057] <hline>

横罫線を指定する。

[属性]

・ x1

横罫線の描画開始位置（ドット単位）を指定する。

・ x2

横罫線の描画終了位置（ドット単位）を指定する。

・ style

(オプション)

罫線の種類を指定する。属性値を図 4 0 A に示す。

[例] XML 文書の例を図 4 0 B に示す。

[0058] <vline-begin>

縦罫線の描画を開始する。

ページモードでは使用できない。

縦罫線は、後述する<vline-end>で終了を指定するまで描画する。本要素は vline-end と一緒に使用する。

[属性]

・ x

縦罫線の描画開始位置（ドット単位）を指定する。

・ style

(オプション)

罫線の種類を指定する。属性値の例を図 4 1 A に示す。

[例] XML 文書の例を図 4 1 B に示す。

[0059] <vline-end>

縦罫線の描画を終了する。

本要素は上記の vline-begin と一緒に使用する。

[属性]

・ x

縦罫線の描画終了位置（ドット単位）を指定する。

・ style

(オプション)

罫線の種類を指定する。属性値の例を図 4 2 A に示す。

[例] XML 文書の例を図 4 2 B に示す。

[0060] <page>

スタンダードモードからページモードへ切り替える。

[<page>の要素]

<page>には図 4 3 A に示す要素を使用できる。

[例] XML 文書の例を図 4 3 B に示す。

[0061] <area>

ページモードの印刷領域を指定する。

絶対原点を基準に、原点、幅、高さを指定して、印字領域を設定する。図 4 4 A に示すように、絶対原点は、印字可能領域の左上のドットである。

印字内容に合わせて印字領域を指定する。印字データが印字領域をはみ出た場合、印字データが途中で切れた印字結果になる。

本要素はpage要素内で使用する。

[属性]

・ x

(初期値: “0”)

横方向の原点をドット単位で指定する。

・ y

(初期値: “0”)

縦方向の原点をドット単位で指定する。

・ width

(初期値: 機種によって異なる)

印字領域の幅をドット単位で指定する。

・ height

(初期値: 機種によって異なる)

印字領域の高さをドット単位で指定する。

印字領域の幅と高さは、印字方向の設定に合わせて確定する。印字データが切れてしまう場合がある。

[例] XML 文書の例を図 4 4 B に示す。

[0062] <direction>

ページモードの印字方向を指定する。

印字方向を指定して、印字領域の回転を行う。印字領域の回転に合わせて

、印字領域の始点が移動する。

本要素はpage要素内で使用する。

[属性]

・ dir

回転方向を指定する。属性値を図 4 5 A に示す。

[例] XML 文書の例を図 4 5 B に示す。

[0063] <position>

ページモードの印字位置を指定する。

印字領域の始点を基準に、印字位置を指定する。印字領域の始点は、印字領域の回転に合わせて移動する。

本要素はpage要素内で使用する。

[属性]

・ x

(初期値: “0”)

横方向位置をドット単位で指定する。

・ y

(初期値: “21”)

縦方向位置をドット単位で指定する。

印字開始位置 (座標) は、印字内容に合わせて指定する。以下を参考にする。

・ 文字列を印字する場合

最初の文字のベースライン左端を指定する。

標準の大きさと、左詰めで印字する場合はオプション能である。高さが2倍の文字を印刷する場合は、yを” 42” 以上に指定する。

・ バーコードを印字する場合

シンボルの左下を指定する。yにバーコードの高さを指定する。

・ グラフィック/ロゴを印字する場合

グラフィックデータの左下を指定する。yにグラフィックデータの高さを

指定する。

- ・ 2次元コードを印字する場合

シンボルの左上を指定する。

[例] XML 文書の例を図 4 6 に示す。

[0064] <line>

ページモードで直線を描画する。

斜線は描画できない。

本要素はpage要素内で使用する。

[属性]

- ・ x1

横罫線の描画開始位置（ドット単位）を指定する。

- ・ y1

縦罫線の描画開始位置（ドット単位）を指定する。

- ・ x2

横罫線の描画終了位置（ドット単位）を指定する。

- ・ y2

縦罫線の描画終了位置（ドット単位）を指定する。罫線の種類を指定する

。

- ・ style

（オプション）

罫線の種類を指定する。属性値を図 4 7 A に示す。

[例] XML 文書の例を図 4 7 B に示す。

[0065] <rectangle>

ページモードで長方形を描画する。

本要素はpage要素内で使用する。

スタンダードモードではhline要素、vline-begin要素、vline-end要素を使用する。

[属性]

- ・ x1

横罫線の描画開始位置（ドット単位）を指定する。

- ・ y1

縦罫線の描画開始位置（ドット単位）を指定する。

- ・ x2

横罫線の描画終了位置（ドット単位）を指定する。

- ・ y2

縦罫線の描画終了位置（ドット単位）を指定する。

- ・ style

（オプション）

罫線の種類を指定する。属性値の例を図 4 8 A に示す。

[例] XML 文書の例を図 4 8 B に示す。

[0066] <cut>

用紙カットを指定する。

スタンダードモードで使用する。

「行の先頭の状態」で実行する。用紙をカットすると「行の先頭の状態」になる。

[属性]

- ・ type

（オプション）

カットの種類を指定する。属性値の例を図 4 9 A に示す。

[例] XML 文書の例を図 4 9 B に示す。

[0067] <pulse>

ドロアーキックコネクターに信号出力を指定する。機種によりブザーを鳴らすことができる。

[属性]

- ・ drawer

（オプション）

ドロアーキックコネクタを指定する。属性値を図 5 0 A に示す。

・ time

(オプション)

ドロアーキック信号のオン時間を指定する。属性値を図 5 0 B に示す。

[例] XML 文書の例を図に示す。

[0068] <sound>

ブザーを鳴らす。

[属性]

・ pattern

(オプション)

ブザーの音色を指定する。属性値を図 5 1 A に示す。

・ repeat

(オプション, 初期値: " 1")

繰り返し回数を指定する。属性値を図 5 1 B に示す。

[例] XML 文書の例を図に示す。

[0069] カスタマーディスプレイのメッセージデータについて説明する。

表示データについては、後述の「カスタマーディスプレイ制御用XML」で説明する。

[0070] <type>display</type>

[リクエスト]

カスタマーディスプレイへ表示データと設定データを送信する。子要素を図 5 2 A に示す。

[例] リクエストの例を図 5 2 B に示す。

[0071] <type>onxmlresult</type>

[レスポンス]

カスタマーディスプレイへ送信した結果を、アプリケーションへ返す。子要素を図 5 3 A に示す。

[例] レスポンスの例を図 5 3 B に示す。

[0072] カスタマーディスプレイを制御するXML文書であるカスタマーディスプレイ制御用XMLについて説明する。

カスタマーディスプレイのメッセージデータについては、上述の「カスタマーディスプレイのメッセージデータ」で説明した通りである。

[0073] <epos-display>

アプリケーションからカスタマーディスプレイへ送信するXML文書である。指定した機能の実行をカスタマーディスプレイにリクエストする。<epos-display>はカスタマーディスプレイの制御に必要な子要素を持つ。

[属性]

・ xmlns

epos-displayの名前を宣言する。子要素を図54Aに示す。

[例] XML文書の例を図54Bに示す。

[0074] <response>

カスタマーディスプレイからアプリケーションへ返すXML文書である。

[属性]

・ success

表示結果を取得する。属性値を図55Aに示す。

・ code

エラーコードを取得する。属性値を図55Bに示す。

・ status

常に"0"がセットされる。

[例] XML文書の例を図55Cに示す。

[0075] <window>

ウィンドウの作成、削除、カレントウィンドウの移動を制御する。

・ 属性number, x, y, width, heightが指定されている場合、指定のウィンドウの作成を行う。ウィンドウを定義する場合は定義済ウィンドウと重ならないように定義する。また、ディスプレイのサイズ（20桁×2行）に収まる大きさを指定する。

- ・属性number destroy="true"が指定されている場合、指定のウィンドウを削除する。

- ・ウィンドウの作成、削除に合致する属性が指定されておらず、numberが指定されている場合、カレントウィンドウを指定のウィンドウに移動する。

[属性]

- ・ number

対象のウィンドウ番号を指定する。属性値を図 5 6 に示す。

- ・ x

ウィンドウ作成時のx座標（1～20）を整数値で指定する。

- ・ y

ウィンドウ作成時のy座標（1～2）を整数値で指定する。

- ・ width

ウィンドウ作成時の幅（1～20）を整数値で指定する。

- ・ height

ウィンドウ作成時の高さ（1～2）を整数値で指定する。

- ・ scrollmode

ウィンドウ作成時、作成ウィンドウのスクロール方法を指定する。属性値を図 5 7 A に示す。

- ・ destroy

numberで指定されたウィンドウを削除するか指定する。属性値を図 5 7 B に示す。

[エラー条件]

エラーの場合、図 5 7 C の値が返る。

[例] XML 文書の例を図 5 7 D に示す。

[0076] <text>

文字列の表示を制御する。

- ・ 属性x及びyが指定されている場合、指定されたカーソル位置から表示を行う。

・ 属性x及びyが指定されていない場合、現在のカーソル位置から表示を行う。

[属性]

・ x

テキスト表示のx座標（1～20）を整数値で指定する。

・ y

テキスト表示のy座標（1～2）を整数値で指定する。

・ reverse

表示文字の明暗反転を指定する。属性値を図58Aに示す。

・ lang

テキスト表示の言語を指定する。省略時は、現在の設定が維持される。属性値を図58Bに示す。

[例] XML文書の例を図58Cに示す。

[0077] <cursor>

カーソル位置および表示設定を制御する。

・ x, yの指定によりディスプレイ内でのカーソル座標を指定する。

・ movetoの指定によりカレントウィンドウ内でのカーソル位置を指定する

。

・ x, y, movetoが同時に指定されている場合、どちらも実施しない。

[属性]

・ x

カーソルのx座標（1～20）を整数値で指定する。

・ y

カーソルのy座標（1～2）を整数値で指定する。

・ moveto

カレントウィンドウ内のカーソル位置を指定する。属性値を図59Aに示す。

・ type

カーソルの表示方法を指定する。省略時は現在の設定が維持される。属性値を図 5 9 B に示す。

[例] XML 文書の例を図 5 9 C に示す。

[0078] <blink>

画面の点滅を制御する。intervalに指定した間隔で明滅を繰り返す。

実際の明滅間隔は、50ms単位で切り上げられる。

例えばintervalが1～50の場合50ms、51～100の場合100msとなる。

[属性]

- ・ interval

点滅の間隔を整数値で指定する。属性値を図 6 0 A に示す。

[例] XML 文書の例を図 6 0 B に示す。

[0079] <brightness>

ディスプレイの輝度を制御する。

[属性]

- ・ value

輝度をパーセントで指定する。属性値を図 6 1 A に示す。

[例] XML 文書の例を図 6 1 B に示す。

[0080] <marquee>

文字列のマーキー表示を制御する。

指定された文字列をuwaitに指定された間隔で1文字ずつ表示する。終端まで表示完了後、rwaitで指定された時間待機し、再度先頭から表示する。

[属性]

- ・ format

マーキーの表示パターンを指定する。属性値を図 6 2 A に示す。

- ・ repeat

表示の繰り返し回数を指定する。属性値を図 6 2 B に示す。

- ・ uwait

1文字表示の待機間隔をミリ秒で指定する。属性値を図 6 2 C に示す。

- ・ rwait

文字列の終端を表示後の待機間隔をミリ秒で指定する。属性値を図 6 2 D に示す。

- ・ lang

表示言語を指定する。省略時は現在の設定が維持される。属性値を図 6 2 E に示す。

[例] XML 文書の例を図 6 2 F に示す。

[0081] <clock>

ディスプレイ画面の最下段右端に時刻を表示する。

表示時刻はプリンター 5 の OS が管理しているローカルタイムである。

[例] XML 文書の例を図 6 3 に示す。

[0082] <clear>

カレントウィンドウの表示を消去する。

ウィンドウが存在しない場合、ディスプレイ全体の表示を消去する。

[例] XML 文書の例を図 6 4 に示す。

[0083] <reset>

ディスプレイの表示を初期化する。初期化により、以下の状態となる。

- ・ 表示文字はすべて消去され、登録されたすべてのウィンドウが破棄される。

- ・ カーソルはデフォルト状態に戻り、ディスプレイ原点に移動する。

- ・ 表示点滅、表示輝度がデフォルト状態に戻る。

[例] XML 文書の例を図 6 5 に示す。

[0084] <command>

任意の ESC/POS コマンドを実行する。

指定するコマンドは 16 進文字列で指定する。

[例] XML 文書の例を図 6 6 に示す。

[0085] POS キーボードのメッセージデータについて説明する。

<type>onkeypress</type>

[レスポンス]

POSキーボードからの入力データを、アプリケーションへ返す。子要素を図67Aに示す。

(受信データの文字列について)

図67Bに示す文字列はエスケープ処理される。

[例] レスポンスの例を図67Cに示す。

[0086] <type>setprefix</type>

POSキーボードからの入力を検出するため、文字列の始まりと判断するキーコードを指定する。本メッセージにより指定されたキーコードが入力された場合、そこからEnterキーが押されるまでの文字列がonstringメッセージとして通知される。例えば、POSキーボードから、バーコードの代替入力を行う場合に使用する。onstringメッセージを止める場合、keycodeのないsetprefixメッセージを送信する。

[リクエスト]

POSキーボードからの入力を検出するための、文字列の始まりを判断するキーコードを指定する。子要素を図68Aに示す。

[例] リクエストの例を図68Bに示す。

[0087] <type>onstring</type>

[レスポンス]

POSキーボードからの入力データを、アプリケーションへ返す。

setprefixで指定したキーコードのいずれかからEnterまでの入力検出されたことを通知する。引数には入力された一連の文字列のほか、検出されたキーコードの情報が取得する。子要素を図69Aに示す。

(受信データの文字列について)

図69Bの文字列はエスケープ処理される。

[例] レスポンスの例を図69Cに示す。

[0088] シリアルデバイスのメッセージデータについて説明する。

<type>sendcommand</type>

[リクエスト]

コマンドを、シリアルデバイスに送信する。子要素を図 7 0 A に示す。

[例] リクエストの例を図 7 0 B に示す。

[0089] <type>oncommandreply</type>

[レスポンス]

シリアルデバイスからのデータを、アプリケーションへ返す。子要素を図 7 1 A に示す。

・ status

属性値を図 7 1 B に示す。

[例] レスポンスの例を図 7 1 C に示す。

[0090] 続いて、デバイス制御スクリプト 5 0 2 について説明する。

まず、プログラミングについて説明する。

図 7 2 は、デバイス制御スクリプトの使用に関する説明図である。

本発明のデバイス制御機能で提供するデバイス制御スクリプト用 API を使うことにより、デバイスのデータ処理のカスタマイズをしたり、新たなデバイスをアプリケーション 3 1 から使用したりすることができる。

図 7 2 に示すように、デバイス制御スクリプト 5 0 2 は、デバイスコネクション 5 0 3 と、クライアントコネクション 5 0 4 とを有する。この図 7 2 では高機能プリンター 5 のデバイスを総称してデバイス 2 0 として示す。

[0091] 上述した open_device メッセージを含む XML データ（リクエスト）が送信されると、プリンター 5 上の Socket I/F 5 0 0 が、XML データを受信する。Socket I/F 5 0 0 は受信したデータをデバイスサービスインターフェイス 5 0 1 に渡す。デバイスサービスインターフェイス 5 0 1 は、open_device メッセージで要求されたデバイスに対応するデバイス制御スクリプトを使用できるようにオブジェクトを生成する。生成されたオブジェクトにより、デバイス 2 0 を制御できる。

[デバイス制御スクリプトのオブジェクト]

デバイス制御スクリプト 5 0 2 には、デバイスサービスインターフェイス

501から図73に示すオブジェクトが渡される。これらクライアント接続 (ClientConnection) オブジェクト、及び、デバイス接続 (DeviceConnection) オブジェクトを使うことで、デバイス制御スクリプト502はアプリケーション31、およびデバイス20と通信することができる。

[0092] デバイス制御スクリプトのオブジェクト使った機能について説明する。

デバイス制御スクリプトAPIを使って以下の機能を使用できる。

- ・アプリケーション31側のデバイスオブジェクトの任意のイベントを呼び出し可能である。

- ・デバイスにデータを送信可能である。

- ・デバイスから発生したデータを受信可能である。

[0093] デバイス制御スクリプトの構成について説明する。

デバイス制御スクリプトは以下の条件を含むようにコーディングされる。

- ・デバイス制御に必要なコードは、1つのファイルにまとめて記述される。

(TMNetWebConfigを使ってデバイスの設定をする際、1つのファイルしか登録できない。)

- ・ファイル名の最初のドット"."までの名称と、コンストラクター名称を同じにする。

[例] ファイル名 : Keyboard_Generic.ver1.0.js→コンストラクターの名称 : Keyboard_Generic

- ・コンストラクターの外部参照のために、exports宣言する。

[例] exports.Keyboard_Generic=Keyboard_Generic

- ・デバイス制御スクリプトには、図74及び図75に示すプロパティを持たせる。コンストラクターで適切な名称を設定される。

DEVICE_TYPEプロパティ (オブジェクト種類 : String)

設定値の一覧を図74に示す。

DEVICE_GROUPプロパティ (オブジェクト種類 : String)

設定値の一覧を図75に示す。

- ・デバイスで発生したデータを受け取るためonDeviceDataメソッドを用意

する。詳細は、後述する「デバイス制御スクリプト名オブジェクト」で説明する。

・アプリケーション 31 側で動作するデバイスオブジェクトのメソッドに対応するメソッドを用意する。詳細は、後述する「任意イベント」で説明する。

デバイス制御スクリプト 502 の構成例を、図 76 に示す。

[0094] デバイス制御スクリプトAPIの一覧を示す。

デバイス制御スクリプトAPIには、次のオブジェクトが用意されている。

- ・ ClientConnection オブジェクト（図 77 にAPIの一覧を示す。）
- ・ DeviceConnection オブジェクト（図 78 にAPIの一覧を示す。）
- ・ デバイス制御スクリプト名オブジェクト（図 79 にAPIの一覧を示す。）

[0095] ClientConnection オブジェクトについて説明する。

このオブジェクトは、デバイス制御スクリプト 502 のコンストラクターの第1パラメーターに渡されるオブジェクトである。

・ send

ブラウザで動作するデバイスオブジェクトにデータを送信する。

[構文]

```
send (event,data) ;
```

[パラメーター]

・ event : オブジェクト種類 : (String)

デバイスオブジェクトのイベント名を設定する。

・ data : オブジェクト種類 : (Object)

デバイスオブジェクトのイベントに渡すデータを指定する。

例を図 80 に示す。この例では、デバイスオブジェクトの onkeypress イベントが呼び出され、onkeypress イベントの data パラメーターを使い、data.keyCode から 49、data.ascii から '1' を取得する。

[0096] DeviceConnection オブジェクトについて説明する。

このオブジェクトは、デバイス制御スクリプト 5 0 2 のコンストラクターの第2パラメーターに渡されるオブジェクトである。

・ send

シリアル通信デバイスにデータを送信する。

[構文]

```
send (data) ;
```

[パラメーター]

・ data : オブジェクト種類 : (Buffer)

デバイスに送信するデータを指定する。

[0097] デバイス制御スクリプト名オブジェクトについて説明する。

・ onDeviceData イベント (HID ドライバーで動作可能な入力デバイス)

このイベントは、キー入力デバイスから検出されたデータを受信するイベントである。

キー入力デバイス用のデバイス制御スクリプト 5 0 2 の場合、この形式でイベントが記述される。

[構文]

```
onDeviceData (event, keycode, ascii) ;
```

[パラメーター]

・ event : オブジェクト種類 : (Number)

キー操作の方向を受け取る。値の一覧を図 8 1 に示す。

・ keycode : オブジェクト種類 : (Number)

キーコードを受け取る。キーコードの例は図 3 A、図 3 B のキーコード一覧に示した。

・ ascii : オブジェクト種類 : (String)

操作されたキーに対応する文字を受け取る。

キーコードに対応する文字がない場合 (F1 キーなど)、undefined になる。

[0098] ・ onDeviceData イベント (シリアル通信デバイス)

このイベントはシリアル通信デバイスからデータを受信するイベントであ

る。

シリアル通信用のデバイス制御スクリプト 5 0 2 の場合、この形式でイベントが記述される。

[構文]

onDeviceData (data) ;

[パラメーター]

・ data : オブジェクト種類 : (Buffer)

シリアル通信デバイスから受信したデータを受け取る。

[0099] ・任意イベント

ブラウザーで動作するデバイスオブジェクトのAPI実行結果を受信するイベントである。

[構文]

callEventで指定した名称 (data) ;

[パラメーター]

・ data : オブジェクト種類 : (Object)

デバイスオブジェクトのcallEventでパラメーター指定したオブジェクトを受け取る。

[0100] プログラムについて説明する。

[概要]

プログラムは、図 8 2 に示すように、バーコードスキャナーでデータを読み取り、印刷する機能を提供する。

[環境]

デバイス

・ プリンター 5

・ バーコードスキャナー 9

実行環境

ここでは次の環境で解説する。

[0101] プログラムの環境構築のフローを図 8 3 A に示す。

1. プリンター5のネットワークを設定する。

システム及びプリンター5のネットワーク設定をする。プリンター5のネットワーク設定は、以下のいずれかの方法により設定できる。

- ・プリンター5で、Windowsのネットワークを設定する。
- ・外部機器（Windowsのコンピュータ）に、TMNet WinConfigをインストールして設定する。

2. プリンター5にバーコードスキャナー9を接続。

プリンター5にバーコードスキャナー9を接続する。

3. バーコードスキャナーの設定。

プリンター5に、バーコードスキャナー9を登録する。登録は、ウェブブラウザから行う。

[0102] [バーコードスキャナーの設定]

デバイスXMLにより制御されるバーコードスキャナー9を、プリンター5に登録する。登録及び設定を、TMNet WebConfigを使って行う。

TMNet WebConfigの画面の例を図94Aに示す。

1. プリンター5にバーコードスキャナー9が接続されているか確認し、プリンター5の電源をオンにする。
2. 設定用のコンピュータでウェブブラウザを起動し、URL（<http://プリンター5のIPアドレス/webconfig/>）を入力する。
3. TMNet WebConfigが起動される。図94Aに（3）で示す「Webサービス設定」-「デバイス管理」-「キー入力デバイス」を選択する。
4. “キー入力デバイス”画面が表示される。図94Aの（4）で示す入力欄に、図94Bに示す項目を設定し、「登録」をクリックする。
5. 登録したバーコードスキャナーが、図94Aの（5）の「登録済みキー入力デバイス」に表示される。

[0103] 以上説明したように、本発明を適用した実施形態のデバイス制御システム100は、アプリケーション31を実行する端末3と、デバイスが接続されるコネクタパネル60を備えたプリンター5とを、ネットワークを介して

接続する。端末3は、アプリケーション31が生成するXMLで記述されたデータをプリンター5に送信する。プリンター5は、端末3から送信されたXMLのデータをSocket I/F500により受信する。プリンター5は、デバイスサービスインターフェイス501及びデバイス制御スクリプト502の機能により、端末3がデータで管理者情報を要求した場合はプリンター5が有する管理者情報を端末3に送信する。また、プリンター5は、端末3がデータでデバイスの制御を要求した場合はデバイスを制御する。このため、ネットワーク接続された端末3により、プリンター5に接続されたデバイスを制御できる。また、端末3は、XMLで記述されたデータを生成するアプリケーションを実行できる装置であれば、制限されない。XMLは汎用性が高いので、様々な装置を端末3として使用できる。また、端末3のアプリケーションの開発が容易である。

[0104] また、プリンター5は、端末3が<admin_info>メッセージを含むデータで管理者情報を要求した場合に、プリンター5が有する管理者名及び／又はロケーションを含むXMLのデータを端末3に返す。

また、プリンター5は、端末3がデータでデバイスのオープンを要求した場合に、デバイスを制御可能な状態にし、端末3がデータでデバイスのクローズを要求した場合に、制御可能なデバイスを制御不能な状態に移行する。

また、アプリケーション31が生成するデータは、プリンター5に接続されたデバイスごとに異なるメッセージデータ、または、デバイスに共通のメッセージデータを含む。

また、プリンター5は、データで要求されたデバイスを制御する場合に、デバイスにデータを送信するDeviceConnectionオブジェクトと、端末3にデータを送信するClientConnectionオブジェクトとを生成する。

[0105] なお、上述した実施の形態は、あくまでも本発明の一態様を示すものであり、本発明の範囲内で任意に変形および応用が可能である。

産業上の利用可能性

[0106] 以上説明したように、本発明は、デバイスを制御するデバイス制御システ

ムに有用であり、POSシステムに用いられるデバイスの制御や印刷装置に適用可能である。

符号の説明

[0107] 2…アプリケーションサーバー、3…端末、5…プリンター、7…ネットワークプリンター、8…カスタマーディスプレイ、9…バーコードスキャナー、11…無線LANアクセスポイント、12…ディスプレイ、13…キャッシュドロワー、14…キーボード、15…USB延長ケーブル、17…無線LANユニット、18…キー入力デバイス、19…シリアル通信デバイス、20…デバイス、31…アプリケーション、51…ロール紙カバー、52…カバーオープンボタン、53…マニュアルカッター、54…カッターカバー、55…ローカルプリンター、56A…リセットボタン、56B…電源スイッチ、57…LED表示部、58…コントロールパネル、59…コネクタカバー、60…コネクタパネル、61…ドロワーキックコネクタ、62…イーサネットコネクタ、63…USBコネクタ、64…VGAコネクタ、65…COMコネクタ、100…デバイス制御システム、500…Socket I/F、501…デバイスサービスインターフェイス、502…デバイス制御スクリプト、503…デバイスコネクション、504…クライアントコネクション。

請求の範囲

- [請求項1] データの入力もしくは出力を行うデバイスと、
マークアップ言語のリクエストの送信及びマークアップ言語のレスポンスを受信するアプリケーションを実行する端末と、
前記デバイスと接続し、前記マークアップ言語のリクエストに基づく前記デバイスの制御及び前記マークアップ言語のレスポンスの送信を行うデバイス制御装置と、
を備えることを特徴とするデバイス制御システム。
- [請求項2] 前記レスポンスは、前記デバイスの制御の結果を示すデータを含む請求項1に記載のデバイス制御システム。
- [請求項3] 前記アプリケーションは、前記デバイス制御装置に通信経路の確立を要求する通信確立メッセージを含む前記リクエストを前記デバイス制御装置に送信し、
前記デバイス制御装置は、前記通信確立メッセージを含む前記リクエストを受信した時、前記リクエストに含まれる前記通信確立メッセージに対する応答メッセージを含む前記レスポンスを前記アプリケーションに送信する請求項1に記載のデバイス制御システム。
- [請求項4] 前記アプリケーションは、前記デバイス制御装置で送信された前記レスポンスに含まれる前記通信確立メッセージに対する応答メッセージに基づいて前記デバイス制御装置との前記通信経路を確立する請求項3に記載のデバイス制御装置。
- [請求項5] 前記アプリケーションは、前記通信経路が確立した後、前記デバイス制御装置に前記デバイス制御装置に接続された前記デバイスとの通信をオープンするオープンメッセージを含む前記リクエストを前記デバイス制御装置に送信する請求項4に記載のデバイス制御装置。
- [請求項6] 前記デバイス制御装置は、前記オープンメッセージを含む前記リクエストを受信した時、前記リクエストに含まれる前記オープンメッセージに対する応答メッセージを含む前記レスポンスを前記アプリケーション

ションに送信する請求項5に記載のデバイス制御システム。

[請求項7] 前記アプリケーションは、前記デバイス制御装置で送信した前記オープンメッセージに対する応答メッセージを含む前記レスポンスを受信し、前記オープンメッセージに対する応答メッセージに基づいて前記デバイスとの通信が可能な状態となる請求項6に記載のデバイス制御システム。

[請求項8] 前記アプリケーションは、前記デバイスとの通信が可能な状態となった後、前記デバイスとの通信をクローズするクローズメッセージを含む前記リクエストを前記デバイス制御装置に送信する請求項7に記載のデバイス制御システム。

[請求項9] 前記デバイス制御装置は、前記クローズメッセージを含む前記リクエストを受信し、
前記クローズメッセージに対応する応答メッセージを含む前記レスポンスを前記アプリケーションに送信する請求項8に記載のデバイス制御システム。

[請求項10] 前記アプリケーションは、前記デバイス制御装置で送信した前記クローズメッセージに対する応答メッセージを含む前記レスポンスを受信し、前記クローズメッセージに対する応答メッセージに基づいて前記デバイスとの通信をクローズする請求項9に記載のデバイス制御システム。

[請求項11] データの入力もしくは出力を行うデバイス、アプリケーションを実行する端末、及び前記デバイスと接続するデバイス制御装置を有し、
前記アプリケーションで生成したマークアップ言語のリクエストを前記デバイス制御装置に送信し、
前記デバイス制御装置で前記リクエストを受信し、
前記マークアップ言語のレスポンスを生成し、
生成した前記レスポンスを前記端末へ送信することを特徴とするデバイス制御システムの制御方法。

- [請求項12] 前記デバイス制御装置で、前記リクエストに含まれる情報を前記デバイスに送信し、
- 前記デバイスは、前記リクエストに含まれる情報に基づいて処理を行い、
- 前記リクエストに含まれる情報に基づいて処理した結果を前記デバイス制御装置に送信し、
- 前記デバイスで送信した前記リクエストに含まれる情報に基づいて処理した結果を含む前記レスポンスを前記端末に送信する請求項11に記載のデバイス制御システムの制御方法。
- [請求項13] 前記アプリケーションで、前記デバイス制御装置に通信経路の確立を要求する通信確立メッセージを含む前記リクエストを前記デバイス制御装置に送信し、
- 前記デバイス制御装置で、前記通信確立メッセージを含む前記リクエストを受信した時、前記リクエストに含まれる前記通信確立メッセージに対する応答メッセージを含む前記レスポンスを前記アプリケーションに送信する請求項1に記載のデバイス制御システムの制御方法。
- [請求項14] 前記アプリケーションで、前記デバイス制御装置で送信された前記レスポンスに含まれる前記通信確立メッセージに対する応答メッセージに基づいて前記デバイス制御装置との前記通信経路を確立する請求項13に記載のデバイス制御装置の制御方法。
- [請求項15] 前記通信経路が確立した後、前記アプリケーションで前記デバイス制御装置に前記デバイス制御装置に接続された前記デバイスとの通信をオープンするオープンメッセージを含む前記リクエストを前記デバイス制御装置に送信する請求項14に記載のデバイス制御システムの制御方法。
- [請求項16] 前記デバイス制御装置で前記オープンメッセージを含む前記リクエストを受信した時、前記リクエストに含まれる前記オープンメッセー

ジに対する応答メッセージを含む前記レスポンスを前記アプリケーションに送信する請求項 15 に記載のデバイス制御システムの制御方法。

[請求項17] 前記アプリケーションで前記デバイス制御装置が送信した前記オープンメッセージに対する応答メッセージを含む前記レスポンスを受信し、

前記オープンメッセージに対する応答メッセージに基づいて前記デバイスとの通信が可能となる請求項 16 に記載のデバイス制御システムの制御方法。

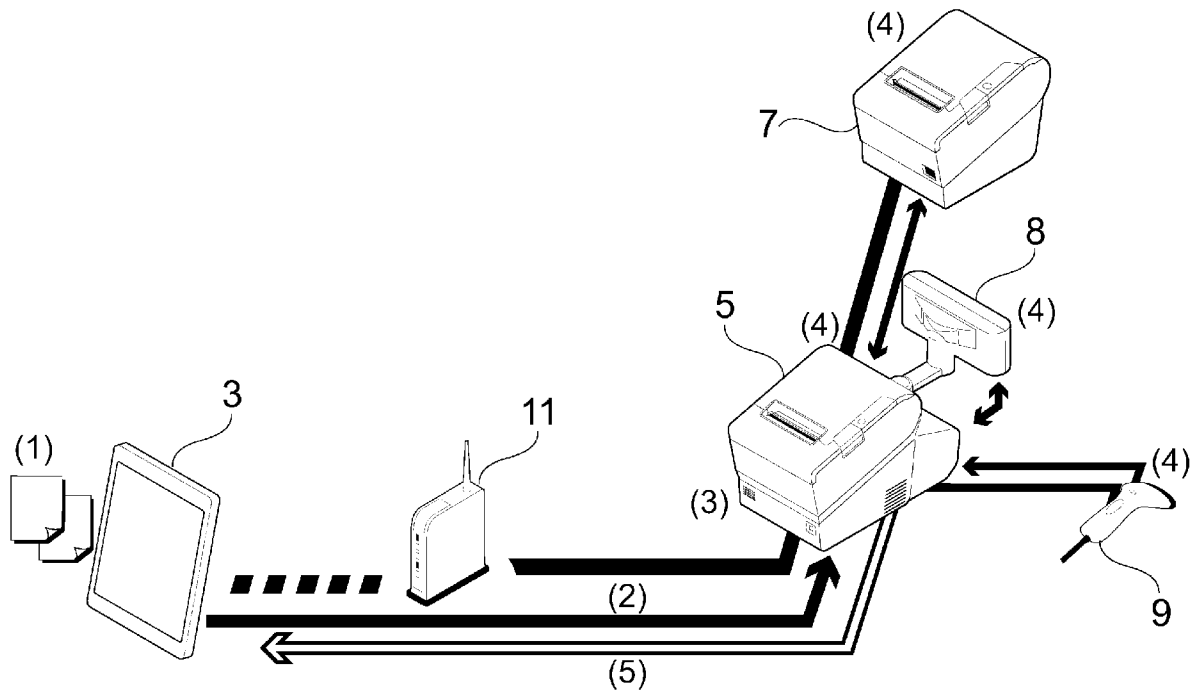
[請求項18] 前記アプリケーションと前記デバイスとの通信が可能になった後に前記アプリケーションを停止させるとき、前記アプリケーションで前記デバイスとの通信をクローズするクローズメッセージを含む前記リクエストを前記デバイス制御装置に送信する請求項 17 に記載のデバイス制御システムの制御方法。

[請求項19] 前記デバイス制御装置で前記クローズメッセージを含む前記リクエストを受信し、
前記クローズメッセージに対応する応答メッセージを含む前記レスポンスを前記アプリケーションに送信する請求項 18 に記載のデバイス制御システムの制御方法。

[請求項20] 前記アプリケーションで前記デバイス制御装置が送信した前記クローズメッセージに対する応答メッセージを含む前記レスポンスを受信し、

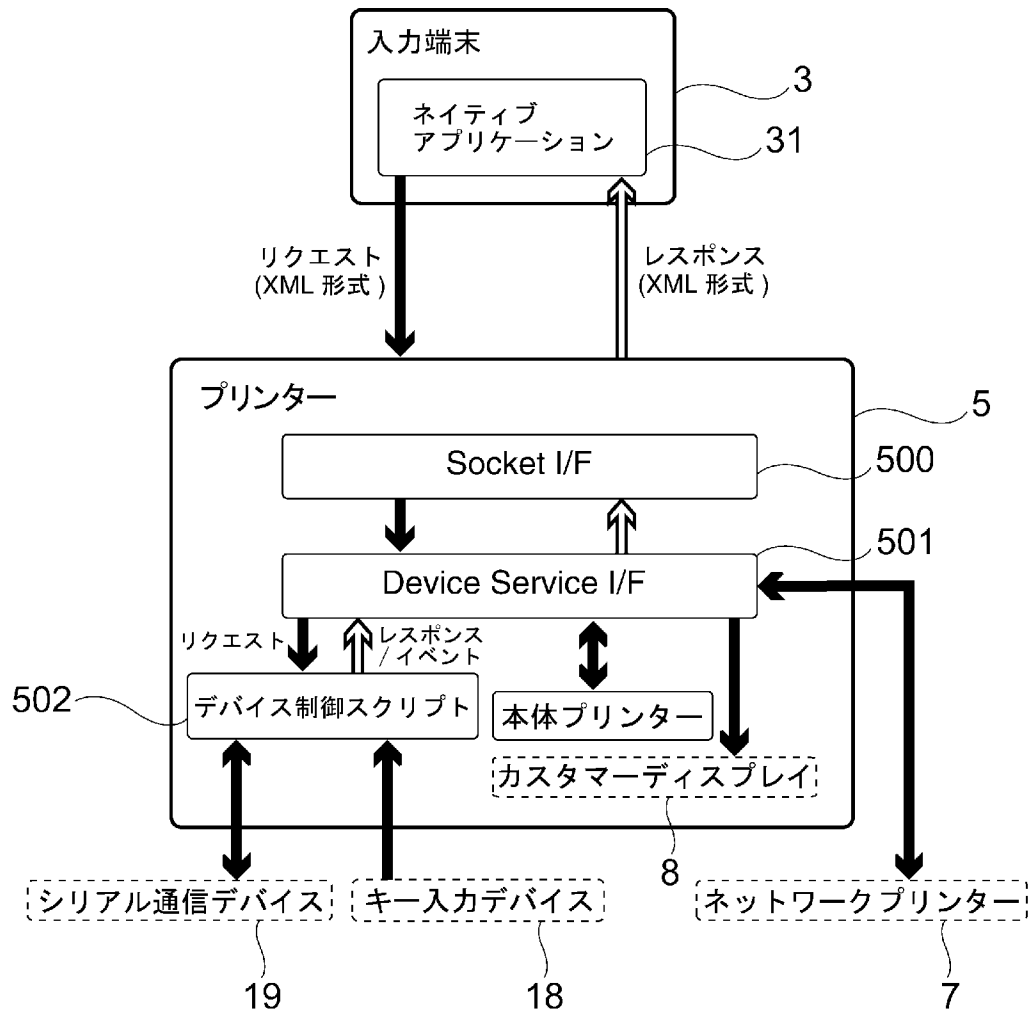
前記クローズメッセージに対する応答メッセージに基づいて前記デバイスとの通信をクローズする請求項 19 に記載のデバイス制御システムの制御方法。

[図1]



[図2]

100



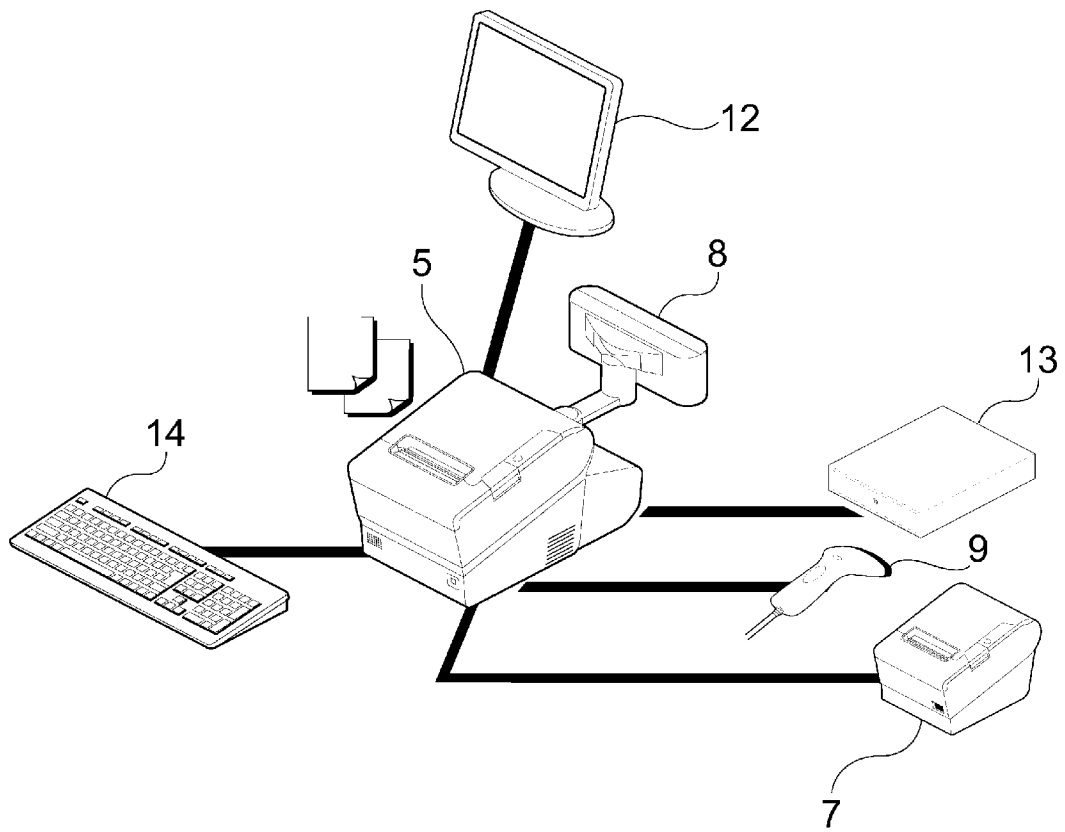
[図3A]

キー	キーコード (10進数)	キー	キーコード (10進数)
Backspace	8	キーボード上の数字	
Tab	9	5	53
Enter	13	6	54
Shift	16	7	55
Ctrl	17	8	56
Alt	18	9	57
Esc	27	キーボード上のアルファベット	
スペース	32	A	65
PgUp	33	B	66
PgDn	34	C	67
End	35	D	68
Home	36	E	69
!	37	F	70
#	38	G	71
i	39	H	72
\$	40	I	73
Insert	45	J	74
Delete	46	K	75
キーボード上の数字		L	76
	0	M	77
	1	N	78
	2	O	79
	3	P	80
	4	Q	81

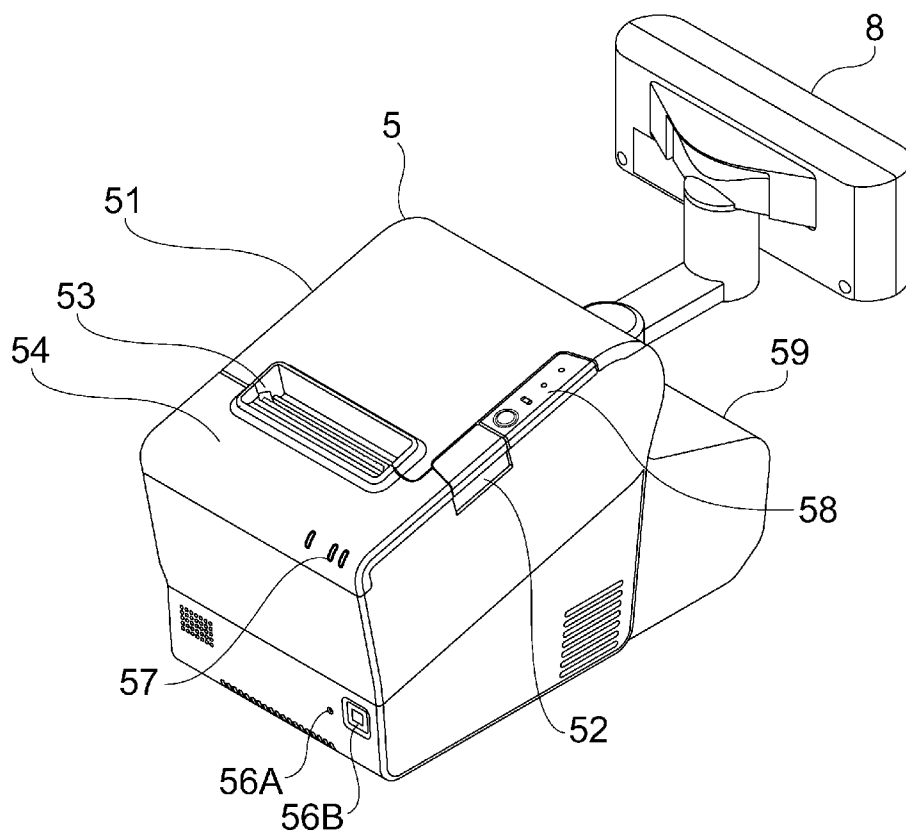
[図3B]

キー	キーコード (10進数)	キー	キーコード (10進数)
キーボード上のアルファベット		ファンクションキー	
R	82	F8	119
S	83	F9	120
T	84	F10	121
U	85	F11	122
V	86	F12	123
W	87	:	186
X	88	;	187
Y	89	,	188
Z	90	-	189
テンキー		.	190
*	106	/	191
+	107	@	192
/	109	[	219
ファンクションキー		\	220
F1	112	]	221
F2	113	^	222
F3	114	Caps Lock	240
F4	115		
F5	116		
F6	117		
F7	118		

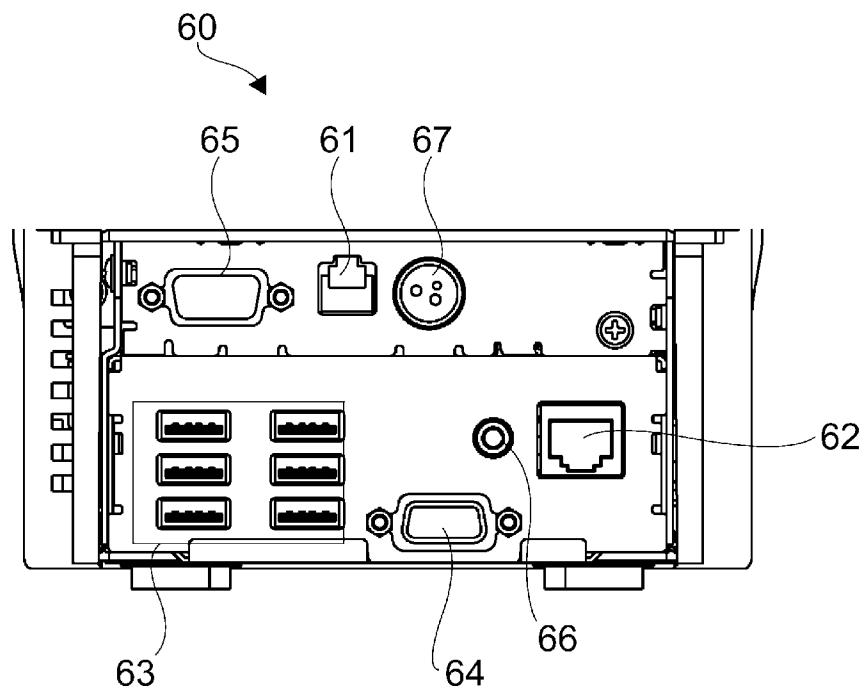
[図4]



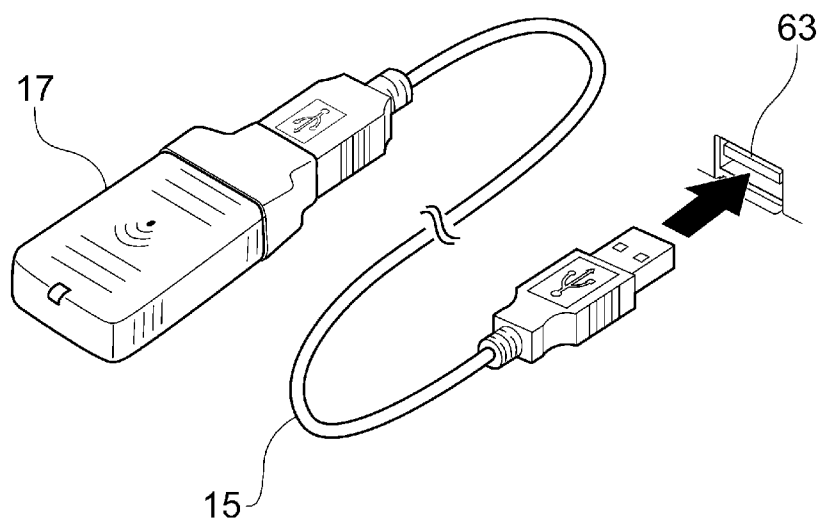
[図5]



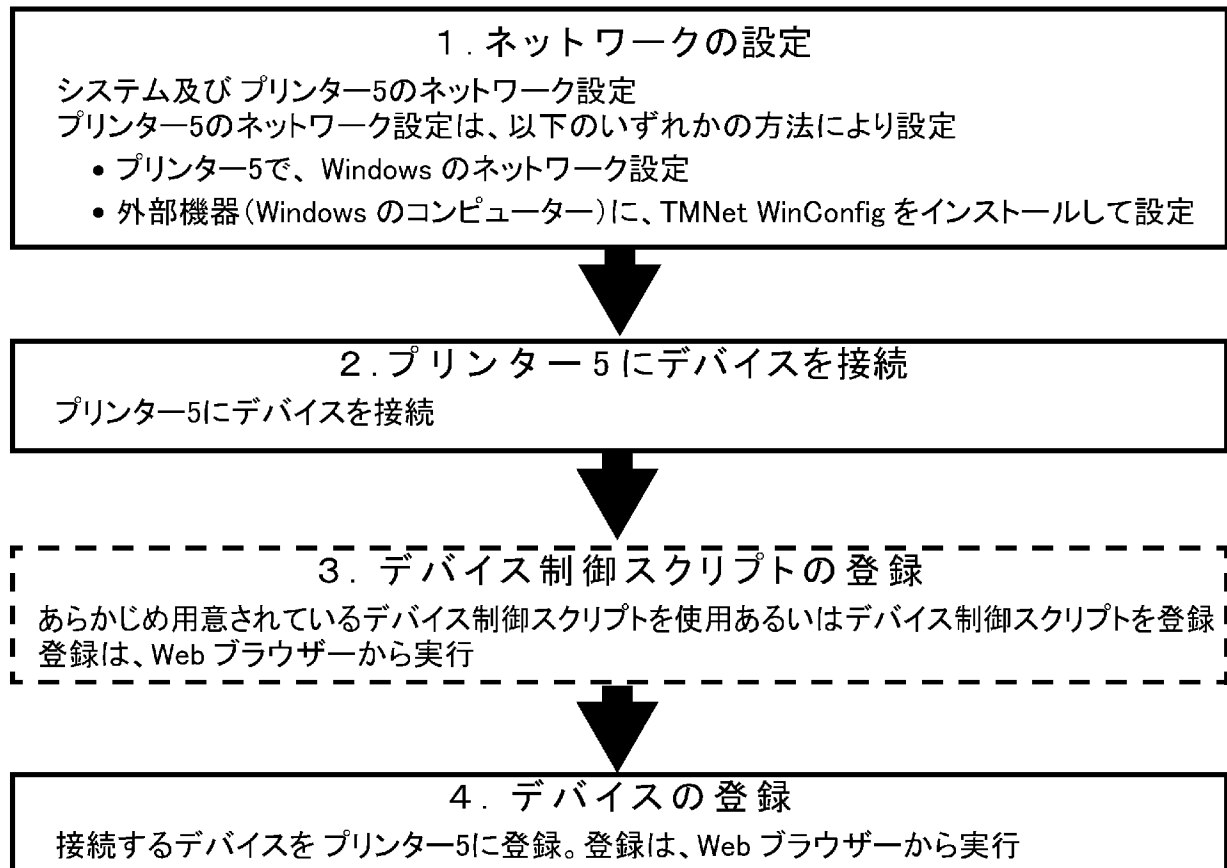
[図6]



[図7]



[図8]



[図9]

**TMNet
WebConfig**

情報
▶ 現在の状態

設定
[Webサービス設定]
起動設定
▶ 起動設定
デバイス管理
デバイス登録
▶ プリンター
▶ ディスプレイ
▶ キー入力デバイス
▶ シリアル通信
▶ その他
制御スクリプト
▶ 登録と削除
Webコンテンツ
▶ 更新設定

制御スクリプト

登録する制御スクリプト

項目名	設定値
制御スクリプト	参照...

登録

登録済み制御スクリプト

キー入力デバイス

デバイスタイプ	制御スクリプト	
type_keyboard	Keyboard_Generic.js	使用中
type_scanner	Scanner_Generic.js	削除

シリアル通信

デバイスタイプ	制御スクリプト	
type_simple_serial	SimpleSerial_Generic.js	使用中

[図10]



情報

Webサービス設定
▶ 現在の状態

設定

[Webサービス設定]

起動設定
▶ 起動設定

デバイス管理
▶ デバイス登録
▶ プリンター
▶ ディスプレイ
▶ キー入力デバイス
▶ シリアル通信
▶ その他

制御スクリプト
▶ 登録と削除

Webコンテンツ
▶ 更新設定

プリンター

登録するプリンター

項目名	設定値
デバイスID	
タイプ	☒ ネットワークプリンター ☐ ローカルプリンター
型番	TM-T88V
IPアドレス	
リトライ間隔	100 ms

登録

登録済みプリンター

デバイスID	型番	IPアドレス	リトライ間隔(ms)		
local_printer	TM-T88V-DT	(ローカルプリンター)	100	テスト印字	削除

[図11]

項目	説明
プリンター	プリンター5で制御するプリンターの設定
ディスプレイ	プリンター5で制御するカスタマーディスプレイの設定
キー入力デバイス	プリンター5で制御するキー入力デバイスの設定
シリアル通信	プリンター5で制御するシリアル通信デバイスの設定
その他	上記に該当しない プリンター5で制御するデバイスの設定

[図12]

項目	説明
デバイスID	制御するプリンターのIDを入力（任意の文字列）
タイプ	“ネットワークプリンター”を選択
型番	制御するプリンターの型番を選択
IPアドレス	デバイスIDごとのプリンターのIPアドレスを設定
リトライ間隔	タイムアウトのリトライ間隔を設定

[図13]

項目	説明
通信設定	通信速度、データビット、パリティを設定
輝度	カスタマーディスプレイの輝度を設定

[図14]

項目	説明
デバイスID	デバイスIDを入力（任意の文字列）
デバイス名	キー入力デバイスのデバイス名を選択
制御スクリプト	キー入力デバイスで使用するデバイス制御スクリプトを選択

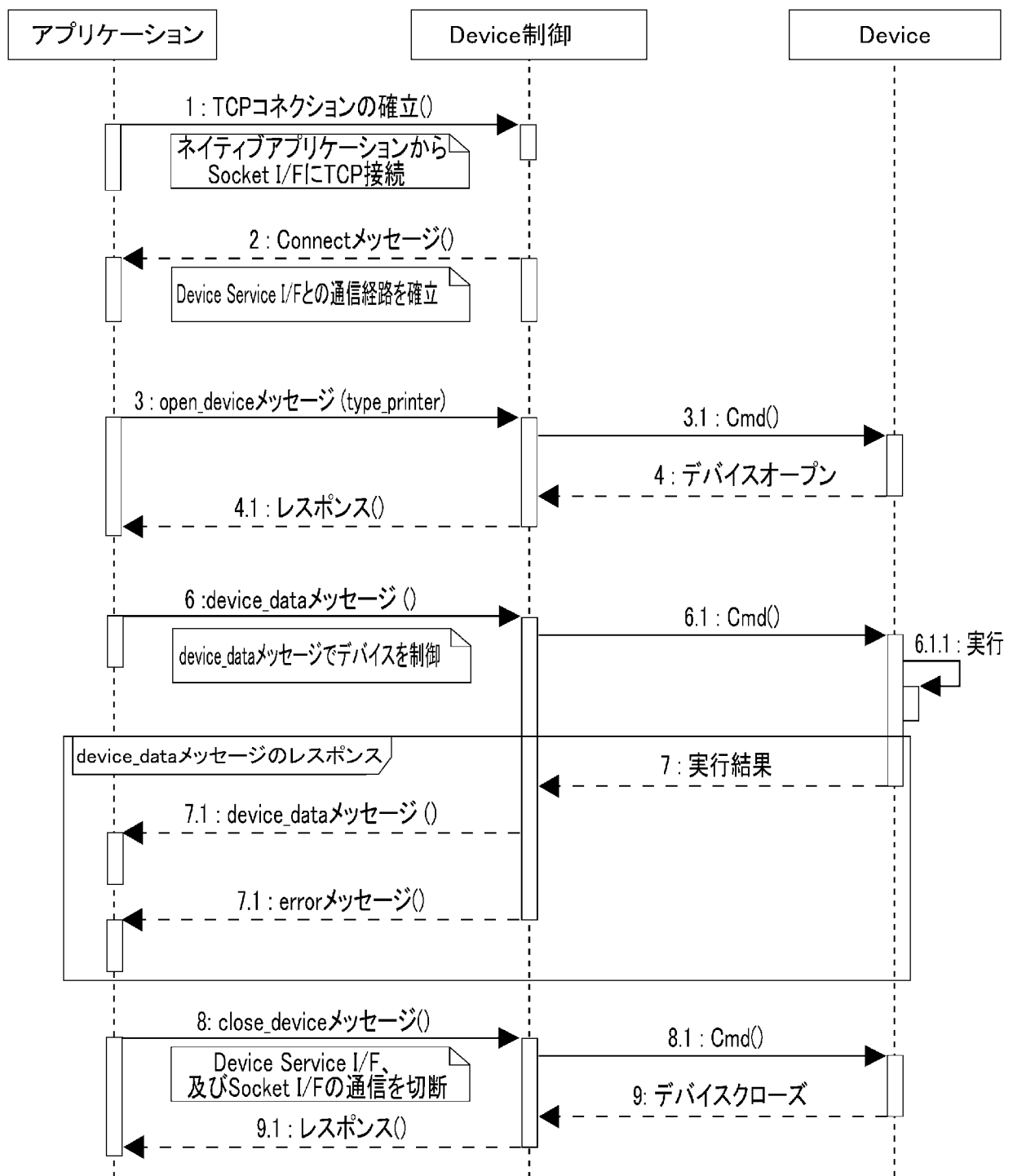
[図15]

項目	説明
デバイス ID	デバイス ID を入力（任意の文字列）
デバイス名	シリアル通信デバイスのデバイス名を選択 製品名、ポートから選択
制御スクリプト	シリアル通信デバイスで使用するデバイス制御スクリプトを選択
通信速度	デバイスの通信速度を設定
データビット	データビットを設定
パリティ	パリティを設定
ストップビット	ストップビットを設定
フロー制御	フロー制御を設定

[図16]

項目	説明
デバイス ID	デバイス ID を入力（任意の文字列）
制御スクリプト	接続したデバイスで使用するデバイス制御スクリプトを選択

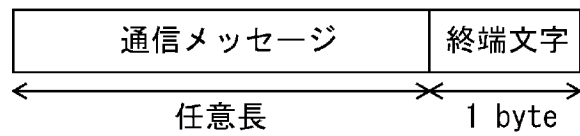
[図17]



[図18]

トランスポート層	ポート番号
TCP	8009

[図19A]



[図19B]

データ名	内容
通信メッセージ	「UTF-8」のXML形式の文字列 任意長
終端文字	NULL文字(“¥0”) 1 byte

[図20A]

アプリケーション	方向	Device	説明
TCP/IP コネクションの確立	➡		接続の確立
	⬅	<connect> </connect>"\0"	
<admin_info></admin_info>"\0"	➡		管理者情報の取得
	⬅	<admin_info> <code>OK</code> <data> <admin_name>ep-admin</admin_name> <location>Counter01</location> </data> </admin_info>"\0"	
<open_device> <device_id>keyboard01</device_id> <data> <type>type_keyboard</type> <buffer>true</buffer> </data> </open_device>"\0"	➡		デバイスのオープン
	⬅	<open_device> <device_id>keyboard01</device_id> <code>OK</code> <data_id>1</data_id>	
<device_data> <device_id>keyboard01</device_id> <data> <type>setprefix</type> <keycode>49</keycode> <keycode>50</keycode> <keycode>51</keycode> </data> </device_data>"\0"	➡		デバイス制御
	⬅	<device_data> <device_id>keyboard01</device_id> <data> <type>onkeypress</type> <keycode>49</keycode> <ascii>a</ascii> </data> <data_id>2</data_id> </device_data>"\0" <device_data> <device_id>keyboard01</device_id> <data> <type>onstring</type> <input>11223344556677</input> <prefix>a</prefix> </data> </device_data>"\0"	

[図20B]

アプリケーション	方向	Device	説明
<code><close_device></code> <code><device_id>keyboard01</device_id></code> <code></close_device>"\0"</code>	➡		デバイスの クローズ
	⬅	<code><close_device></code> <code><device_id>keyboard01</device_id></code> <code><code>OK</code></code> <code></close_device>"\0"</code>	
接続の切断			

[図21]

```

<data>
  <type>setprefix</type>
  <keycodes array="true">49</keycodes>
  <keycodes>50</keycodes>
  <keycodes>51</keycodes>
  <keycodes>52</keycodes>
</data>

```

[図22]

メッセージ		説明
通信経路	<connect>	ePOS-Device Service I/Fとの通信経路を確立
管理情報	<admin_info>	管理者、設置場所情報の取得
デバイスの オープン / クローズ	<open_device>	デバイスとの通信をオープン
	<close_device>	デバイスとの通信をクローズ
デバイス制御データ の送信	<device_data>	デバイス制御をリクエスト
	<device_data>	デバイス制御のレスポンス
エラー通知	<error>	エラー発生時のエラー内容の通知

[図23]

デバイスの種類	<type> 要素の値	説明
バーコード スキャナー	ondata	バーコードスキャナーからのスキャンデータ <レスポンス>
プリンター	print	プリンターへの印刷データと設定データ <リクエスト>
	onxmlresult	プリンターからのデータ <レスポンス>
カスタマーディスプレイ	display	カスタマーディスプレイへの表示データと設定データ <リクエスト>
	onxmlresult	カスタマーディスプレイからのデータ <レスポンス>
POSキーボード	onkeypress	POSキーボードからの入力文字列 <レスポンス>
	setprefix	POSキーボードからの入力を検出するため、文字列の始まりと判断するキーコードを指定 <リクエスト>
	onstring	POSキーボードからの入力文字列 <レスポンス>
シリアルデバイス	sendcommand	シリアルデバイスへのコマンドの送信 <リクエスト>
	oncommandreply	シリアルデバイスからのデータ <レスポンス>

[図24A]

子要素	データ型	説明
code	string	管理者情報の取得結果 詳細は下記 code を参照
data	string	管理者情報 取得に成功した場合に返信 詳細はdataを参照

[図24B]

要素の値	説明
"OK"	取得成功
"PARAM_ERROR"	パラメーターエラー発生
"SYSTEM_ERROR"	システムエラー発生

[図24C]

子要素	データ型	説明
admin_name	string	管理者情報
location	string	設置場所情報

[図24D]

置換前文字列	置換後
& (アンパサンド)	&
> (大なり記号)	>
< (小なり記号)	<
" (二重引用符)	"
' (単一引用符)	'

[図24E]

正常時

```

<admin_info>
  <code>OK</code>
  <data>
    <admin_name>管理者A</admin_name>
    <location>カウンター1 &amp; カウンター2</location>
  </data>
</admin_info>

```

異常時

```

<admin_info>
  <code>PARAM_ERROR</code>
</admin_info>

```

[図25A]

子要素	データ型	説明
device_id	string	TMNet WebConfigで指定済みのデバイスID カスタマーディスプレイは"local_display"
data	string	詳細はdataを参照

[図25B]

要素の値	説明
"type_scanner"	バーコードスキャナー
"type_keyboard"	POSキーボード
"type_printer"	プリンター
"type_display"	カスタマーディスプレイ
"type_simple_serial"	シリアル通信デバイス

[図25C]

```

<open_device>
  <device_id>scanner_001</device_id>
  <data>
    <type>type_scanner</type>
  </data>
</open_device>

```

[図25D]

子要素	データ型	説明
device_id	string	リクエストで指定したデバイス ID
code	string	処理結果 詳細はcodeを参照

[図25E]

要素の値	説明
"OK"	処理に成功
"DEVICE_NOT_FOUND"	指定されたデバイスが存在しない
"DEVICE_IN_USE"	指定されたデバイスが使用中
"DEVICE_OPEN_ERROR"	指定されたデバイスのオープンに失敗
"DRIVER_ERROR"	デバイス制御スクリプトでエラー発生
"SYSTEM_ERROR"	システムエラー発生
"DEVICE_CATEGORY_INVALID"	指定されたデバイスの種類が違う
"COMMAND_PARAM_ILLEGAL"	パラメーターエラー発生

[図25F]

正常時

```

<open_device>
  <code>OK</code>
  <data_id>scanner_001</device_id>
</open_device>

```

異常時

```

<open_device>
  <code>DEVICE_NOT_FOUND</code>
  <device_id>scanner_001</device_id>
</open_device>

```

[図26A]

子要素	データ型	説明
device_id	string	TMNet WebConfigで指定済みのデバイスID カスタマーディスプレイは"local_display"

[図26B]

```
<close_device>
  <device_id>local_display</device_id>
</close_device>
```

[図26C]

子要素	データ型	説明
device_id	string	リクエストで指定したデバイスID
code	string	処理結果 詳細はcodeを参照

[図26D]

要素の値	説明
"OK"	処理に成功
"DEVICE_NOT_FOUND"	指定されたデバイスが存在しない
"DEVICE_NOT_OPEN"	指定されたデバイスIDがオープンされていない
"DEVICE_CLOSE_ERROR"	指定されたデバイスのクローズに失敗
"SYSTEM_ERROR"	システムエラー発生
"COMMAND_PARAM_ILLEGAL"	パラメーターエラー発生

[図26E]

正常時

```
<close_device>
  <device_id>local_display</device_id>
  <code>OK</code>
</close_device>
```

異常時

```
<close_device>
  <device_id>scanner_001</device_id>
  <code>DEVICE_NOT_FOUND</code>
</close_device>
```


[図27A]

子要素	データ型	説明
sequence	string	送信エラー時に、リクエストメッセージを特定するための任意の数値 リクエストメッセージに指定すると、エラー発生時にerrorメッセージ のsequenceにセット
device_id	string	オープン済みのデバイスID
data	string	デバイス制御のデータ 内容はデバイスの種類ごとに定義

[図27B]

```

<device_data>
  <sequence>123</sequence>
  <device_id>poskeyboard001</device_id>
  <data>
    <type>setprefix</type>
    <keycodes array="true">49</keycodes>
    <keycodes>50</keycodes>
    <keycodes>51</keycodes>
    <keycodes>52</keycodes>
  </data>
</device_data>

```

[図27C]

子要素	データ型	説明
sequence	string	0 固定
device_id	string	データが発生したデバイスID
data	string	発生したデータ 内容はデバイスの種類ごとに定義

[図27D]

```

<device_data>
  <sequence>0</sequence>
  <device_id>poskeyboard001</device_id>
  <data>
    <type>onstring</type>
    <input>&2398749238429</input>
    <prefix>49</prefix>
  </data>
</device_data>

```

[図28A]

子要素	データ型	説明
sequence	string	エラーの原因となるメッセージを特定するID リクエストで指定したsequenceと同じ値
device_id	string	エラーの発生したデバイスID デバイス制御メッセージでのエラーの場合に指定
code	string	発生したエラーをコードで表示
data	string	DRIVER_ERROR発生時、ドライバに応じたエラーの詳細が指定

[図28B]

- device_data でのエラー時

要素の値	説明
"DEVICE_NOT_FOUND"	指定されたデバイスIDが存在しない
"DEVICE_NOT_OPEN"	指定されたデバイスがオープンされていない
"SEND_ERROR"	該当の物理デバイスへのデータ送信でエラーが発生
"DRIVER_ERROR"	ドライバでエラーを検出
"PARAM_ERROR"	パラメータエラー発生
"FUNCTION_NOT_FOUND"	指定された関数がドライバに存在しない

- その他エラー時

要素の値	説明
"SYSTEM_ERROR"	システムエラー発生

[図28C]

```

<error>
  <sequence>123</sequence>
  <device_id>scanner002</device_id>
  <code>DEVICE_NOT_OPEN</code>
  <data></data>
/errro>

```

[図29A]

子要素	データ型	説明
input	string	受信したスキャンデータの文字列

[図29B]

置換前文字列	置換後
& (アンパサンド)	&
> (大なり記号)	>
< (小なり記号)	<
" (二重引用符)	"
' (単一引用符)	'

[図29C]

```
<data>
  <type>ondata</type>
  <input>F12345678909123</input>
</data>
```

[図30A]

子要素	データ型	説明
timeout	int	リクエストのタイムアウト時間をミリ秒で指定 1000 ~ 60000(整数)
printdata	string	印刷するデータを指定

[図30B]

```
<data>
  <type>print</type>
  <timeout>10000</timeout>
  <printdata>
    <!-- プリンター制御用 XML -->
    <epos-print xmlns=
      .
      .
    </epos-print>
  </printdata>
</data>
```

[図31A]

子要素	データ型	説明
resultdata	string	送信したデータの結果

[図31B]

```

<data>
  <type>onxmlresult</type>
  <resultdata>
    <response success=      />
  </resultdata>
</data>

```

[図32A]

子要素	機能	プリンターの印刷モード	
		スタンダード モード	ページ モード
<text>	文字の印字	○	○
<feed>	紙送り	○	○
<image>	ラスターイメージの印字	○	○
<logo>	NV ロゴの印字	○	○
<barcode>	バーコードの印字	○	○
<symbol>	2次元コードの印字	○	○
<hline>	横罫線の印字	○	△
<vline-begin>	縦罫線の開始	○	△
<vline-end>	縦罫線の終了	○	△
<page>	ページモード	○	△
<area>	ページモード印字領域の設定	△	○
<direction>	ページモード印字方向の設定	△	○
<position>	ページモード印字位置の設定	△	○
<line>	ページモード直線の描画	△	○
<rectangle>	ページモード長方形の描画	△	○
<cut>	用紙カット	○	△
<pulse>	ドロアキック	○	△
<sound>	ブザーの鳴動	○	△
<command>	コマンドの挿入	○	○

[図32B]

```

<!-- 子要素として text タグを追加した例 -->
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <text>Hello world!!</text>
</epos-print>

```

[図33A]

属性の値	説明
"true" または "1"	表示成功
"false" または "0"	表示失敗

[図33B]

属性の値	説明
"EPTR_AUTOMATIC"	自動復帰エラー発生
"EPTR_COVER_OPEN"	カバーオープンエラー発生
"EPTR_CUTTER"	オートカッターエラー発生
"EPTR_MECHANICAL"	メカニカルエラー発生
"EPTR_REC_EMPTY"	ロール紙エンド検出器に用紙なし
"EPTR_UNRECOVERABLE"	復帰不可能エラー発生
"SchemaError"	要求ドキュメントの構文に誤りがある
"DeviceNotFound"	デバイスIDで指定したプリンターが存在しない
"PrintSystemError"	印刷システムにエラーが発生
"EX_BADPORT"	通信ポートで異常が検出
"EX_TIMEOUT"	印刷タイムアウトが発生

[図33C]

属性の値	説明
"0x00000001"	TMプリンター無応答
"0x00000002"	印刷完了
"0x00000004"	ドロアキックコネクタ3番ピンの状態="H"
"0x00000008"	オフライン状態
"0x00000020"	カバーが開いている
"0x00000040"	紙送りスイッチによる紙送信中
"0x00000100"	オンライン復帰待ち中
"0x00000200"	紙送りスイッチが押下中
"0x00000400"	メカニカルエラー発生
"0x00000800"	オートカッターエラー発生
"0x00002000"	復帰不可能エラー発生
"0x00004000"	自動復帰エラー発生
"0x00020000"	ロール紙ニアエンド検出器に用紙なし
"0x00080000"	ロール紙エンド検出器に用紙なし
"0x01000000"	ブザーが鳴動中 (対応機器のみ)
"0x80000000"	スプーラーの停止 (未使用)

[図33D]

```

<!-- Printのプリンターのstatus -->
<!-- 成功時 -->
  <response success="true" code="" status="" />
<!-- 印字中に用紙が無くなり、印刷が失敗した場合 -->
<response xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print"
  success="false" code="EPTR_REC_EMPTY" status="252641308" />

```

[図34A]

機能（記号）	実体参照
&	&
'	'
>	>
<	<
"	"
水平タブ (HT)		 または 	
改行 (LF)	
 または

[図34B]

属性の値	説明
"en"（初期値）	英語
"ja"	日本語

[図34C]

属性の値	説明
"font_a"（初期値）	フォントA
"font_b"	フォントB
"font_c"	フォントC

[図34D]

属性の値	説明
"true" または "1"	スムージングを指定
"false" または "0"（初期値）	スムージングを解除

[図34E]

属性の値	説明
"true" または "1"	横倍角を指定
"false" または "0"（初期値）	横倍角を解除

[図34F]

属性の値	説明
"true" または "1"	縦倍角を指定
"false" または "0" (初期値)	縦倍角を解除

[図34G]

属性の値	説明
"1" ~ "8" (初期値 : "1")	横方向倍率

[図34H]

属性の値	説明
"1" ~ "8" (初期値 : "1")	縦方向倍率

[図34I]

属性の値	説明
"true" または "1"	白黒反転文字を指定
"false" または "0" (初期値)	白黒反転文字を解除

[図34J]

属性の値	説明
"true" または "1"	アンダーラインを指定
"false" または "0" (初期値)	アンダーラインを解除

[図34K]

属性の値	説明
"true" または "1"	強調印字を指定
"false" または "0" (初期値)	強調印字を解除

[図34L]

属性の値	説明
"none"	非印字
"color_1" (初期値)	第 1 色
"color_2"	第 2 色
"color_3"	第 3 色
"color_4"	第 4 色

[図34M]

属性の値	説明
"left" (初期値)	左揃え
"center"	中央揃え
"right"	右揃え

[図34N]

属性の値	説明
"true" または "1"	倒立印字を指定
"false" または "0" (初期値)	倒立印字を解除

[図34P]

項目	設定値
言語	日本語
スムージング	有効
位置揃え	中央揃え
フォント	フォントA
倍角	倍角
アンダーライン	有効

```

<text lang="ja" />
<text smooth="true" />
<text align="center" />
<text font="font_a" />
<text dw="true" dh="true" />
<text ul="true" />
<text>Hello, World!&#10;</text>

```

[図35]

```

<!-- 紙送りを3行する -->
<feed line="3" />

```

```

<!-- 文字列を印字して改行する -->
<text>Hello</text><feed />
<text>World</text><feed />

```


[図36A]

属性の値	説明
"none"	非印字
"color_1" (初期値)	第 1 色
"color_2"	第 2 色
"color_3"	第 3 色
"color_4"	第 4 色

[図36B]

属性の値	説明
"left" (初期値)	左揃え
"center"	中央揃え
"right"	右揃え

[図36C]

属性の値	説明
"mono" (初期値)	モノクロ(2階調)
"gray16"	多階調(16階調)

[図36D]

```
<!-- 幅8ドット、高さ8ドットで塗りつぶすラスターイメージを印字する -->
<image width="8" height="8">//////////8=</image>
```

[図37A]

属性の値	説明
"left" (初期値)	左揃え
"center"	中央揃え
"right"	右揃え

[図37B]

```
<!-- キーコード 1 が 48、キーコード 2 が 48 に登録されている NV ロゴを印字する -->
<logo key1="48" key2="48" />
```

[図38A]

種類	バーコードデータの指定方法
UPC-A	11桁の数字を指定した場合、チェックデジットを付加 12桁の数字を指定した場合、12桁目をチェックデジットとして処理
UPC-E	最初の桁に0を指定 2～6桁目にメーカーコードを指定 7～11桁目にアイテムコードを右詰めで指定 11桁の数字を指定した場合、チェックデジットを付加 12桁の数字を指定した場合、12桁目をチェックデジットとして処理
EAN13	12桁の数字を指定した場合、チェックデジットを付加
JAN13	13桁の数字を指定した場合、13桁目をチェックデジットとして処理
EAN8	7桁の数字を指定した場合、チェックデジットを付加
JAN8	8桁の数字を指定した場合、8桁目をチェックデジットとして処理
CODE39	先頭の文字が*の場合、この文字をスタートキャラクターとして処理 それ以外の場合、スタートキャラクターを付加
ITF	スタートコードおよびストップコードを付加
CODABAR	スタートキャラクター(A～D, a～d)を指定 ストップキャラクター(A～D, a～d)を指定

[図38B]

種類	バーコードデータの指定方法														
CODE93	スタートキャラクターおよびストップキャラクターを付加														
CODE128	スタートキャラクター(CODE A, CODE B, CODE C)を指定 ストップキャラクターを付加 チェックデジットを計算して付加 以下の文字をエンコードするには、文字 { で始まる2文字を指定 FNC1 : {1 FNC2 : {2 FNC3 : {3 FNC4 : {4 CODE A : {A CODE B : {B CODE C : {C SHIFT : {S { : {{														
GS1-128	スタートキャラクター、FNC1、チェックデジット、ストップキャラクターを付加 アプリケーション識別子(AI)とそれに続くデータのチェックデジットを計算して付加するには、チェックデジットの位置に文字 * を指定 アプリケーション識別子(AI)を括弧で囲むことが可能 括弧はHRIの印字文字として使用 アプリケーション識別子(AI)とデータの間空白を挿入することが可能 空白はHRIの印字文字として使用 以下の文字をエンコードするには、文字 { で始まる2文字を指定 <table> <tr> <td>文字指定する</td><td>文字</td></tr> <tr> <td>FNC1</td><td>{1</td></tr> <tr> <td>FNC3</td><td>{3</td></tr> <tr> <td>(</td><td>{{</td></tr> <tr> <td>)</td><td>{}</td></tr> <tr> <td>*</td><td>{*</td></tr> <tr> <td>{</td><td>{{</td></tr> </table>	文字指定する	文字	FNC1	{1	FNC3	{3	(	{{	)	{}	*	{*	{	{{
文字指定する	文字														
FNC1	{1														
FNC3	{3														
(	{{														
)	{}														
*	{*														
{	{{														
GS1 DataBar Omnidirectional,	アプリケーション識別子(AI)とチェックデジットを除く13桁の商品識別番号(GTIN)を指定														
GS1 DataBar Truncated															
GS1 DataBar Limited															

[図38C]

種類	バーコードデータの指定方法
BARCODE_GS1_DataBar Expanded	アプリケーション識別子(AI)を括弧で囲むことが可能 括弧はHRIの印字文字として使用 以下の文字をエンコードするには、文字 { で始まる2文字を指定 FNC1 : {1 (: {{) : {}

[図38D]

文字列	説明
\xnn	コントロールコード
\\	バックスラッシュ

[図38E]

属性の値	説明
"upc_a"	UPC-A
"upc_e"	UPC-E
"ean13"	EAN13
"jan13"	JAN13
"ean8"	EAN8
"jan8"	JAN8
"code39"	CODE39
"itf"	ITF
"codabar"	CODABAR
"code93"	CODE93
"code128"	CODE128
"gs1_128"	GS1-128
"gs1_databar_omnidirectional"	GS1 DataBar Omnidirectional
"gs1_databar_truncated"	GS1 DataBar Truncated
"gs1_databar_limited"	GS1 DataBar Limited
"gs1_databar_expanded"	GS1 Databar Expanded

[図38F]

属性の値	説明
"none" (初期値)	印字しない
"above"	バーコードの上
"below"	バーコードの下
"both"	バーコードの上と下の両方

[図38G]

属性の値	説明
"font_a" (初期値)	フォントA
"font_b"	フォントB
"font_c"	フォントC

[図38H]

属性の値	説明
"left" (初期値)	左揃え
"center"	中央揃え
"right"	右揃え

[図38I]

属性の値	説明
"true" または "1"	倒立印字を指定
"false" または "0" (初期値)	倒立印字を解除

[図38J]

<!-- 各種バーコードを指定する -->

```

<barcode type="upc_a" width="2" height="64" hri="below">01234567890</barcode>
<barcode type="upc_e">01234500005</barcode>
<barcode type="ean13">201234567890</barcode>
<barcode type="jan13">201234567890</barcode>
<barcode type="ean8">2012345</barcode>
<barcode type="jan8">2012345</barcode>
<barcode type="code39">ABCDE</barcode>
<barcode type="itf">012345</barcode>
<barcode type="codabar">A012345A</barcode>
<barcode type="code93">ABCDE</barcode>
<barcode type="code128">{Babcde</barcode>
<barcode type="gs1_128">(01)201234567890*</barcode>
<barcode type="gs1_databar_omnidirectional">0201234567890</barcode>
<barcode type="gs1_databar_truncated">0201234567890</barcode>
<barcode type="gs1_databar_limited">0201234567890</barcode>
<barcode type="gs1_databar_expanded">(01)2012345678903</barcode>

```

[図39A]

種類	説明
PDF417	文字列をUTF-8に変換後、エスケープシーケンスの処理を行い、エンコードデータ領域の最大コードワード数は928個、1段あたりのデータ領域の最大コードワード数は30個、最大段数は90段
QR Code	文字列をシフトJISに変換後、エスケープシーケンスの処理を行い、データの種別を以下の中から選択してエンコード 数字データ : 0 ~ 9 英数字データ : 0 ~ 9, A ~ Z, スペース, \$, %, *, +, -, ., /, : 漢字データ : シフト JIS値 8ビットバイトデータ : 0x00 ~ 0xff

[図39B]

種類	説明
MaxiCode	文字列を UTF-8に変換後、エスケープシーケンスの処理を行い、エンコードモード2およびモード3の場合、最初のデータが[]>¥x1e01¥x1dy (yyは2桁の数字) の場合、これをメッセージヘッダーとして処理し、次のデータからプライマリメッセージとして処理 それ以外の場合、最初のデータからプライマリメッセージとして処理 プライマリメッセージを以下の形式で指定 ●モード 2 の場合 郵便コード : (1 ~ 9 桁の数字) GS : (\x1d) ISO 国名コード : (1 ~ 3 桁の数字) GS : (\x1d) サービスクラスコード (1 ~ 3 桁の数字) ●モード 3 の場合 郵便コード : (1 ~ 6 個のコードセット A で変換可能なデータ) GS : (\x1d) ISO 国名コード : (1 ~ 3 桁の数字) GS : (\x1d) サービスクラスコード (1 ~ 3 桁の数字)
GS1 DataBar Stacked	文字列を UTF-8に変換後、エスケープシーケンスの処理を行い、エンコード
GS1 DataBar Stacked Omnidirectional	アプリケーション識別子(AI) とチェックデジットを除く13 桁の商品識別番号 (GTIN) を指定
GS1 DataBar Expanded Stacked	文字列を UTF-8に変換後、エスケープシーケンスの処理を行い、エンコードアプリケーション識別子(AI)を括弧で囲むことが可能 括弧はHRIの印字文字として使用 以下の文字をエンコードするには、文字 { で始まる2文字を指定 FNC1 : {1 (: {() : {}

[図39C]

文字列	説明
\xnn	コントロールコード
\\	バックスラッシュ

[図39D]

属性の値	種類
"pdf417_standard"	Standard PDF417
"pdf417_truncated"	Truncated PDF417
"qrcode_model_1"	QR Code Model 1
"qrcode_model_2"	QR Code Model 2
"maxicode_mode_2"	MaxiCode Mode 2
"maxicode_mode_3"	MaxiCode Mode 3
"maxicode_mode_4"	MaxiCode Mode 4
"maxicode_mode_5"	MaxiCode Mode 5
"maxicode_mode_6"	MaxiCode Mode 6
"gs1_databar_stacked"	GS1 DataBar Stacked
"gs1_databar_stacked_omnidirectional"	GS1 DataBar Stacked Omnidirectional
"gs1_databar_expanded_stacked"	GS1 DataBar Expanded Stacked

[図39E]

属性の値	説明
"level_0"	PDF417エラー訂正レベル 0
"level_1" (初期値)	PDF417エラー訂正レベル 1
"level_2"	PDF417エラー訂正レベル 2
"level_3"	PDF417エラー訂正レベル 3
"level_4"	PDF417エラー訂正レベル 4
"level_5"	PDF417エラー訂正レベル 5
"level_6"	PDF417エラー訂正レベル 6
"level_7"	PDF417エラー訂正レベル 7
"level_8"	PDF417エラー訂正レベル 8
"level_l"	QR Code エラー訂正レベル L
"level_m"	QR Code エラー訂正レベル M
"level_q"	QR Code エラー訂正レベル Q
"level_h"	QR Code エラー訂正レベル H
"default"	既存の設定を保持

[図39F]

2次元コード	有効な値の範囲	初期値
PDF417	"2" ~ "8"	width="3"
QR Code	"1" ~ "16"	width="3"
MaxiCode	無視	
GS1 DataBar	"2" ~ "8"	width="2"

[図39G]

2次元コードの種類	有効な値の範囲	初期値
PDF417	"2" ~ "8"	height="3"
QR Code	無視	
MaxiCode		
2次元 GS1 Databar		

[図39H]

2次元コードの種類	初期値	説明
PDF417	size="0"	1段あたりのコードワード数を指定
QR Code	無視	
MaxiCode		
GS1 Databar	機種によって異なる	バーコードの最大幅を指定

[図39I]

属性の値	説明
"left" (初期値)	左揃え
"center"	中央揃え
"right"	右揃え

[図39J]

属性の値	説明
"true" または "1"	倒立印字を指定
"false" または "0" (初期値)	倒立印字を解除

[図39K]

```

<!-- 各種 2 次元コードを指定する -->
<symbol type="pdf417_standard">ABCDE</symbol>
<symbol type="qrcode_model_2" level="level_q">ABCDE</symbol>
<symbol type="maxicode_mode_2">908063840\x1d850\x1d001\x1d\x04</symbol>
<symbol type="gs1_databar_stacked">0201234567890</symbol>
<symbol type="gs1_databar_stacked_omnidirectional">0201234567890</symbol>
<symbol type="gs1_databar_expanded_stacked">(01)02012345678903</symbol>

```

[図40A]

属性の値	説明
"thin"	実線: 細
"medium"	実線: 中太
"thick"	実線: 太
"thin_double"	二重線: 細
"medium_double"	二重線: 中太
"thick_double"	二重線: 太

[図40B]

```

<!-- 左端 100ドットから 200ドット、左端 400ドットから 500ドットの横罫線を二重線で描画する -->
<hline x1="100" x2="200" style="thin_double" />
<hline x1="400" x2="500" style="thin_double" />

```


[図41A]

属性の値	説明
"thin"	実線: 細
"medium"	実線: 中太
"thick"	実線: 太
"thin_double"	二重線: 細
"medium_double"	二重線: 中太
"thick_double"	二重線: 太

[図41B]

```

<!-- 左端 100ドットと 200ドットの位置から細線で描画する -->
<vline-begin x="100" />
<vline-begin x="200" />
<feed unit="100" />
<vline-end x="100" />
<vline-end x="200" />

```

[図42A]

属性の値	説明
"thin"	実線: 細
"medium"	実線: 中太
"thick"	実線: 太
"thin_double"	二重線: 細
"medium_double"	二重線: 中太
"thick_double"	二重線: 太

[図42B]

```

<!-- 左端 100ドットと 200ドットの位置まで細線で描画する -->
<vline-begin x="100" />
<vline-begin x="200" />
<feed unit="100" />
<vline-end x="100" />
<vline-end x="200" />

```

[図43A]

要素	説明
<text>	文字の印字
<feed>	紙送り
<image>	ラスターイメージの印字
<logo>	NV ロゴの印字
<barcode>	バーコードの印字
<symbol>	2次元コードの印字
<area>	印字領域の設定
<direction>	印字方向の設定
<position>	印字位置の設定
<line>	直線の描画
<rectangle>	長方形の描画
<command>	コマンドの挿入

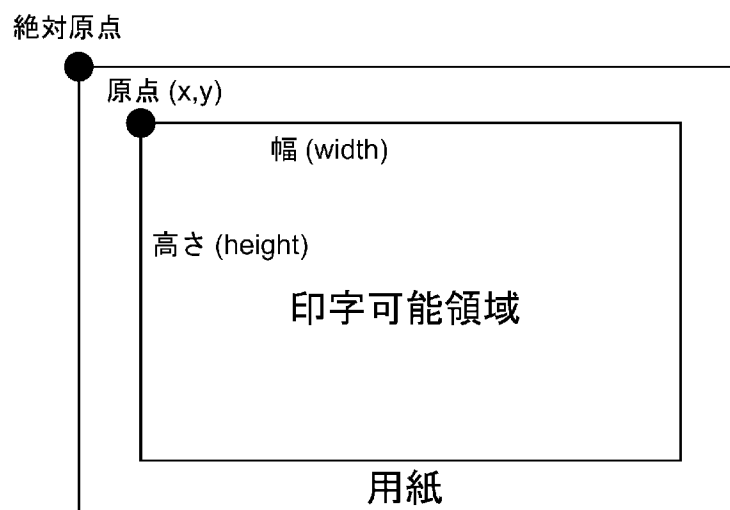
[図43B]

```

<!-- ページモードで文字「ABCDE」を印字する -->
<page>
<text>ABCDE</text>
</page>

```

[図44A]



[図44B]

```

<!-- 原点(100,50)、幅200ドット、高さ30ドットの印字領域を指定して、文字「ABCDE」を印字する -->
<page>
  <area x="100" y="50" width="200" height="30" />
  <text>ABCDE</text>
</page>

```

[図45A]

属性の値	説明
"left_to_right" (初期値)	左→右 (回転しない。左上を始点に右方向へ印字)
"bottom_to_top"	下→上 (反時計回り 90 度回転。左下を始点に上方向へ印字)
"right_to_left"	右→左 (180 度回転。右下を始点に左方向へ印字)
"top_to_bottom"	上→下 (時計回り 90 度回転。右上を始点に下方向へ印字)

[図45B]

```

<!-- 時計回りに90度回転させて文字「ABCDE」を印字する -->
<page>
  <direction dir="top_to_bottom" />
  <text>ABCDE</text>
</page>

```

[図46]

```

<!-- area要素で指定したエリア内の印字開始位置を(50,30)に指定して、文字「ABCDE」を印字する -->
<page>
  <area x="100" y="50" width="200" height="100" />
  <position x="50" y="30" />
  <text>ABCDE</text>
</page>

```

[図47A]

属性の値	説明
"thin"	実線:細
"medium"	実線:中太
"thick"	実線:太
"thin_double"	二重線:細
"medium_double"	二重線:中太
"thick_double"	二重線:太

[図47B]

```

<!-- 開始位置(100,0)、終了位置(500,0)を原点とする直線を細い実線で描画する -->
<page>
<line x1="100" y1="0" x2="500" y2="0" style="thin" />
</page>

```

[図48A]

属性の値	説明
"thin"	実線:細
"medium"	実線:中太
"thick"	実線:太
"thin_double"	二重線:細
"medium_double"	二重線:中太
"thick_double"	二重線:太

[図48B]

```

<!-- 開始位置(100,0)、終了位置(500,200)を頂点とする四角形を細い二重線で描画する -->
<page>
<rectangle x1="100" y1="0" x2="500" y2="200" style="thin_double" />
</page>

```

[図49A]

属性の値	説明
"no_feed"	フィード無しカット (紙送りを実行せずにカット)
"feed"	フィードカット (カット位置まで紙送りを実行しカット)
"reserve"	カット予約 (後に続く印字を実行しカット位置到達後カット)

[図49B]

```

<!-- フィードカットする -->
<cut type="feed" />

```

[図50A]

属性の値	説明
"drawer_1" (初期値)	ドロアーキックコネクタ 2 番ピン
"drawer_2"	ドロアーキックコネクタ 5 番ピン

[図50B]

属性の値	説明
"pulse_100" (初期値)	100 ミリ秒の信号
"pulse_200"	200 ミリ秒の信号
"pulse_300"	300 ミリ秒の信号
"pulse_400"	400 ミリ秒の信号
"pulse_500"	500 ミリ秒の信号

[図50C]

```

<!-- ドロアーキックコネクタ 2 番ピンに 100 ミリ秒のパルス信号を出力する -->
<pulse drawer="drawer_1" time="pulse_100" />

```

[図51A]

属性の値	説明
"none"	停止
"pattern_a" (初期値)	パターン A
"pattern_b"	パターン B
"pattern_c"	パターン C
"pattern_d"	パターン D
"pattern_e"	パターン E
"error"	エラー鳴動パターン
"paper_end"	紙無し鳴動パターン

[図51B]

属性の値	説明
"0"	無限
"1" ~ "255"	指定回数繰り返す

[図51C]

```
<!-- パターン A を 3 回鳴らす -->
<sound pattern="pattern_a" repeat="3" />
```

[図52A]

子要素	データ型	説明
timeout	int	リクエストタイムアウト時間をミリ秒で指定 1000 ~ 60000(整数)
displaydata	string	カスタマーディスプレイ制御用XMLを指定

[図52B]

```
<data>
  <type>display</type>
  <timeout>10000</timeout>
  <displaydata>
    <!-- カスタマーディスプレイ制御用 XML -->
    <epos-display xmlns=
      .
      .
    </epos-display>
  </displaydata>
</data>
```

[図53A]

子要素	データ型	説明
resultdata	string	カスタマーディスプレイに送信したデータのレスポンスを受信

[図53B]

```

<data>
  <type>onxmlresult</type>
  <resultdata>
    <response success="true" code="" status="" />
  </resultdata>
</data>

```

[図54A]

子要素	機能
<window>	ウィンドウの設定
<text>	文字の表示
<cursor>	カーソルの表示
<blink>	画面の点滅を設定
<brightness>	ディスプレイの輝度を設定
<marquee>	文字列のマーキー表示を設定
<clock>	時刻の表示
<clear>	カレントウィンドウの表示の消去
<reset>	ディスプレイ表示の初期化
<command>	ESC/POSコマンドの実行

[図54B]

```

<!-- 子要素として text タグを追加した例 -->
<epos-display xmlns="http://www.epson-pos.com/schemas/2012/09/epos-display">
  <text>Hello world!!</text>
</epos-display>

```

[図55A]

属性の値	説明
"true" または "1"	表示成功
"false" または "0"	表示失敗

[図55B]

属性の値	説明
"EDSP_NOT_FOUND"	デバイスが見つからない
"EDSP_NOT_OPEN"	デバイスのオープンに失敗
"EDSP_INVALID_WINDOW"	未登録のウィンドウが指定
"EDSP_SYSTEM_ERROR"	予期しないエラーが発生
"EDSP_SCHEMA_ERROR"	XML 構造が間違っている
"EX_BADPORT"	デバイスとの内部通信エラーが発生
"EX_TIMEOUT"	デバイスとの通信でタイムアウトエラーが発生
"EX_INVALID_VALUE"	パラメータ不正を検出

[図55C]

```

<!-- 成功時 -->
  <response success="true" code="" status=" "/>
<!-- 失敗時 -->
  <response success="false" code="EDSP_NOT_FOUND" status=" "/>

```

[図56]

属性の値	説明
"1"	ウィンドウ 1
"2"	ウィンドウ 2
"3"	ウィンドウ 3
"4"	ウィンドウ 4

[図57A]

属性の値	説明
"overwrite"(初期値)	表示位置が上段右端位置にあるとき文字表示すると、表示位置を下段左端に移動 下段右端にあるとき文字表示すると表示位置を上段左端に移動
"v_scroll"	表示位置が上段右端位置にあるとき文字表示すると、表示位置を下段左端に移動 下段右端にあるとき文字表示するとすでに表示してある下段の表示文字を上段にスクロールし、下段の表示をクリア
"h_scroll"	表示位置が右端位置にあるとき文字表示すると、カーソルのある行にすでに表示している全ての表示文字は1文字分左へスクロールし、新たな表示文字を右端に表示

[図57B]

属性の値	説明
"true" または "1"	削除を実施
"false" または "0"(初期値)	削除を実施しない

[図57C]

エラー値	説明
"EX_INVALID_VALUE"	ウィンドウ作成時、x, y, width, heightが既に定義済みのウィンドウの範囲にかかる場合
"EDSP_INVALID_WINDOW"	ウィンドウ移動時、指定されたウィンドウが存在しない場合

[図57D]

```

<!-- 左上から幅10、高さ2、縦スクロールのウィンドウ1を定義する -->
<window number="1" x="1" y="1" width="10" height="2" scrollmode="v_scroll"/>
<!-- 中央から幅10、高さ2、縦スクロールのウィンドウ2を定義する -->
<window number="2" x="11" y="1" width="10" height="2" scrollmode="v_scroll"/>
<!-- カレントウィンドウをウィンドウ1に移動する -->
<window number="1"/>
<!-- ウィンドウ 2 を削除する -->
<window number="2" destroy="true"/>

```

[図58A]

属性の値	説明
"true" または "1"	明暗反転
"false" または "0"(初期値)	明暗反転させない

[図58B]

属性の値	説明
"en"(初期値)	英語
"ja"	日本語

[図58C]

```

<!-- 左上から文字列を表示する -->
<text x="1" y="1" lang="ja">ニジソ \100</text>
<!-- 現在のカーソル位置から明暗反転状態で文字列を表示する -->
<text reverse="true">Welcome!!</text>

```

```

<!-- カレントウィンドウをウィンドウ2に切り替えて表示 -->
<window number="2"/>
<text x="1" y="1" lang="ja"> ニジソ $1</text>

```

[図59A]

属性の値	説明
"top_left"	左上に移動
"top_right"	右上に移動
"bottom_left"	左下に移動
"bottom_right"	右下に移動

[図59B]

属性の値	説明
"none"	カーソルを表示しない
"underline"	アンダーライン形状で表示

[図59C]

```

<!-- カーソルをディスプレイ左上に移動する -->
<cursor x="1" y="1"/>
<!-- カーソルをカレントウィンドウ内の左上に移動する -->
<cursor moveto="top_left" type="underline"/>

```

[図60A]

属性の値	説明
"0"	点滅を終了し連続点灯状態にする
"1" ~ "12700"	点滅の間隔のミリ秒
"12701" ~ "12750"	表示内容を保持した状態で消灯する

[図60B]

```

<!-- 画面の点滅間隔を 500ms にする -->
<blink interval="500"/>
<!-- 画面の点滅を解除する -->
<blink interval="0"/>
<!-- 画面を消灯する。表示内容は維持する -->
<blink interval="12750"/>
<cursor moveto="top_left" type="underline"/>

```

[図61A]

属性の値	説明
"20"	輝度 20%
"40"	輝度 40%
"60"	輝度 60%
"100"	輝度 100%

[図61B]

```

<!-- 輝度を 20% に設定する -->
<brightness value="20"/>
<!-- 輝度を 40% に設定する -->
<brightness value="40"/>
<!-- 輝度を 60% に設定する -->
<brightness value="60"/>
<!-- 輝度を 100% に設定する -->
<brightness value="100"/>

```

[図62A]

属性の値	説明
"walk" (初期値)	ウインドウ右端から文字列を表示
"place"	ウインドウ左端から文字列を表示

[図62B]

属性の値	説明
"0" (初期値)	無限に繰り返す
"1" ~ "127" (整数)	指定回数繰り返す

[図62C]

属性の値	説明
"0" ~ "2000" (整数)	1 文字表示ごとの待機間隔

[図62D]

属性の値	説明
"100" ~ "2000" (整数)	文字列終端を表示後の待機時間

[図62E]

属性の値	説明
"en" (初期値)	英語
"ja"	日本語

[図62F]

```

<!-- 画面右から200ms/文字でマーキー表示し、1000ms間隔で繰り返す -->
<marquee uwait="200" rwait="1000">イラッシャイマセ</marquee>
<!-- 画面左から200ms/文字でマーキー表示し、1000ms間隔で繰り返す -->
<marquee uwait="200" rwait="1000" format="place">Welcome!!</marquee>

```

[図63]

```
<clock/>
```

[図64]

```
<clear/>
```

[図65]

```
<reset/>
```

[図66]

```
<command>01025AEF3B405C</command>
```

[図67A]

子要素	データ型	説明
keycode	int	POSキーボードで入力されたキーコードの10進数を受け取り
ascii	string	キーコードにより生成された文字を受け取り シフトの押下状態による大文字、記号などに変換された文字が設定

[図67B]

置換前文字列	置換後
& (アンパサンド)	&
> (大なり記号)	>
< (小なり記号)	<
" (二重引用符)	"
' (単一引用符)	'

[図67C]

```
<data>
  <type>onkeypress</type>
  <keycode>49</keycode>
  <ascii>a</ascii>
</data>
```

[図68A]

子要素	データ型	説明
keycode	int	文字列の始まりと判断するキーコードの10進数を指定

[図68B]

```
<data>
  <type>setprefix</type>
  <keycodes array="true">49</keycodes>
  <keycodes>50</keycodes>
  <keycodes>51</keycodes>
  <keycodes>52</keycodes>
</data>
```

[図69A]

子要素	データ型	説明
input	string	入力文字列を受け取り 文字列の始まりと判断するためのキーと、Enterキーは含まない
prefix	int	文字列の始まりと判断したキーコードを受け取り

[図69B]

置換前文字列	置換後
& (アンパサンド)	&
> (大なり記号)	>
< (小なり記号)	<
" (二重引用符)	"
' (単一引用符)	'

[図69C]

```

<data>
  <type>onstring</type>
  <input>&amp;2398749238429</input>
  <prefix>49</prefix>
</data>

```

[図70A]

子要素	データ型	説明
command	string	コマンド列の16進表現文字列

[図70B]

```

<data>
  <type>sendcommand</type>
  <command>1B123344FF1F5D3C</command>
</data>

```

[図71A]

子要素	データ型	説明
status	string	データ受信の状態
data	string	データ列の16進表現文字列

[図71B]

要素の値	説明
"SUCCESS"	データ受信成功

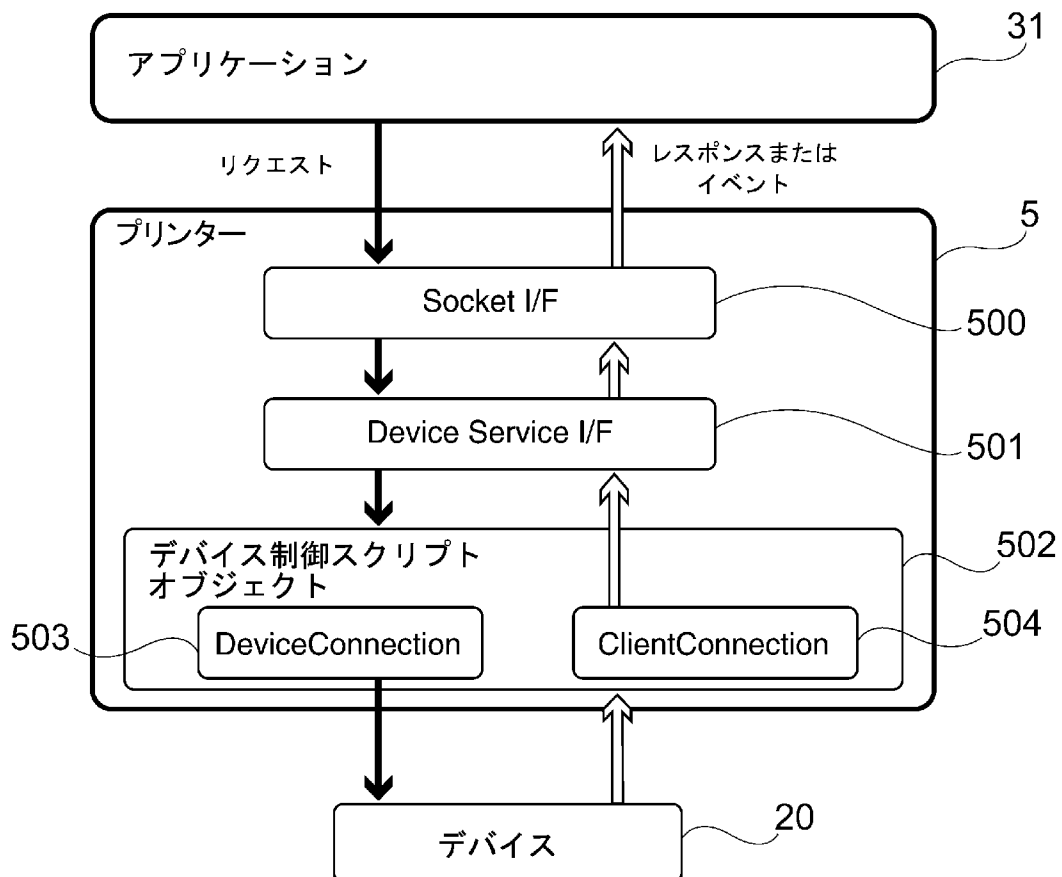
[図71C]

```

<data>
  <type>oncommandreply</type>
  <status>SUCCESS</status>
  <data>1B123344FF1F5D3C</data>
</data>

```

[図72]



[図73]

オブジェクト	説明
ClientConnection	アプリケーション側のデバイスオブジェクトにデータ送信を行うためのオブジェクト
DeviceConnection	デバイスとのデータ送受信を行うオブジェクト

[図74]

設定値	説明
type_keyboard	キーボードデバイスを使用する場合に指定
type_scanner	バーコードスキャナーを使用する場合に指定
type_simple_serial	単純なシリアル通信を行う場合に指定

[図75]

設定値	説明
group_hid	HIDドライバーで動作可能な入力デバイスを使用する場合に指定
group_serial	シリアル通信デバイスを使用する場合に指定
group_other	その他のデバイスを使用する場合に指定

[図76]

```

exports.Keyboard_Generic = Keyboard_Generic;           // exports 宣言

function Keyboard_Generic(clientConn, deviceConn){      // 2つの引数を持ち、ファイル名
                                                         // と同名
    this.DEVICE_TYPE = 'type_keyboard';                // DEVICE_TYPE プロパティを
                                                         // 持つ
    this.DEVICE_GROUP = 'group_hid';                   // DEVICE_GROUP プロパティ
                                                         // を持つ

    this.clientConn = clientConn;
    this.deviceConn = deviceConn;
    .....
    .....
}

Keyboard_Generic.prototype = {
    onDeviceData : function(event, keycode, ascii){...}, // onDeviceData メソッドを持つ
    setprefix : function(data){...}                     // デバイスオブジェクト対応した
                                                         // メソッドがある
}

```

[図77]

要素	API	説明
メソッド		
	send	ブラウザで動作するデバイスオブジェクトにデータを送信

[図78]

要素	API	説明
メソッド		
	send	シリアル通信デバイスにデータを送信

[図79]

要素	API	説明
イベント		
	onDeviceData	キー入力受信イベント
	onDeviceData	シリアル通信デバイスからデータ受信イベント
	任意イベント	ブラウザで動作するデバイスオブジェクトのAPI 実行結果受信イベント

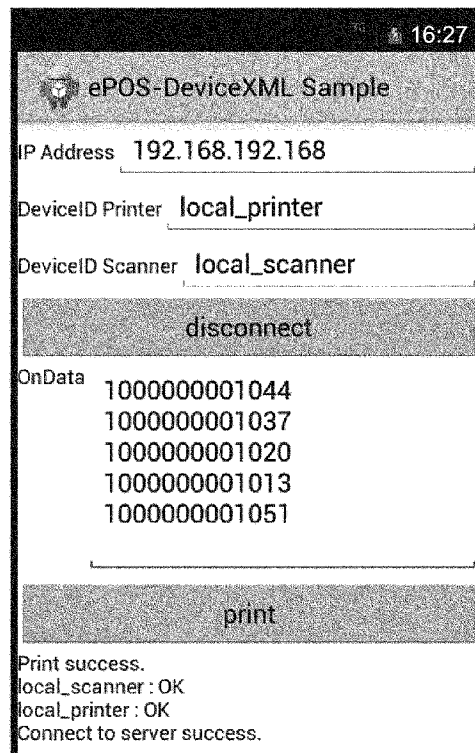
[図80]

```
data = {'keycode' : 49, 'ascii' : '1'};  
clientConn.send('onkeypress', data)
```

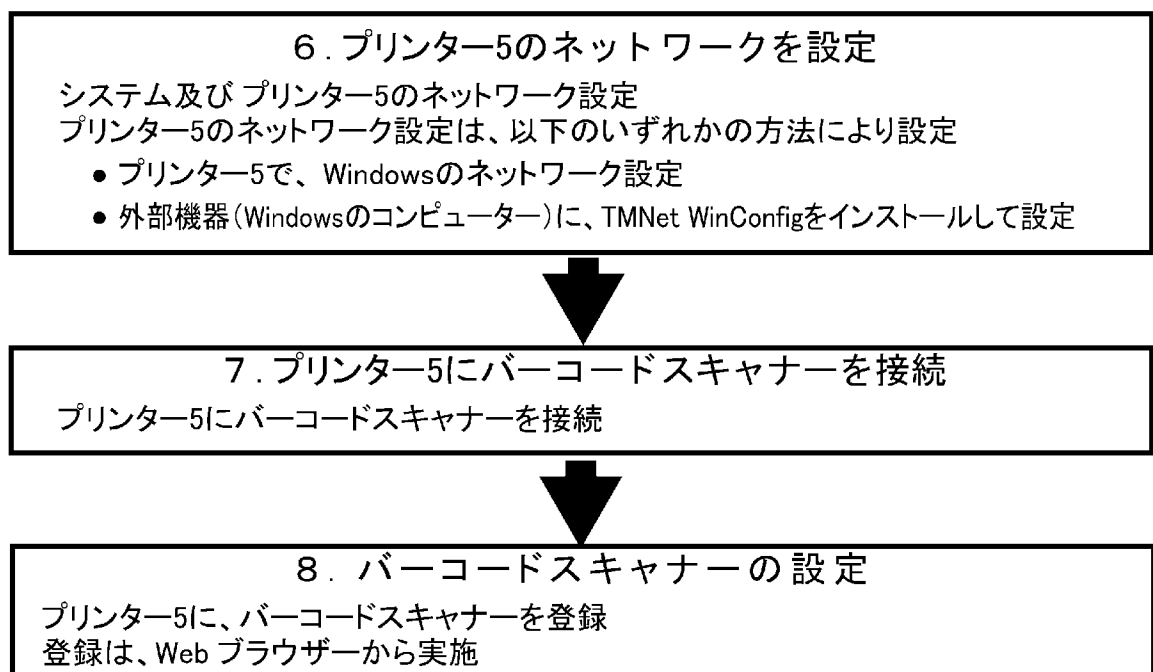
[図81]

値	説明
1	キーダウン
2	キーアップ

[図82]



[図83]



[図84A]

TMNet WebConfig

情報

Webサービス設定

現在の状態

設定

{Webサービス設定}

起動設定

起動設定

デバイス管理

デバイス登録

コントローラ

ディスプレイ

③ キー入力デバイス

シリアル通信

その他

制御スクリプト

登録と削除

ウェブコンテンツ登録設定

キー入力デバイス

登録するキー入力デバイス

項目名	設定値
デバイスID	
デバイス名	unknown : unknown(vid:0566 pid:3002)
制御スクリプト	Keyboard_Generic.js

登録

⑤ 登録済みキー入力デバイス

デバイスID	デバイス名	VID / PID	制御スクリプト		
local_scanner	DENSO WAVE INC.: USB Device HCS6	076D / 0001	Scanner_Generic.js	動作テスト	削除

[図84B]

項目名	設定値
デバイス ID	local_scanner
デバイス名	リストから該当するデバイスを選択
制御スクリプト	Scanner_Generic.js

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2013/007633

A. CLASSIFICATION OF SUBJECT MATTER

G06F13/00(2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F13/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2014
Kokai Jitsuyo Shinan Koho	1971-2014	Toroku Jitsuyo Shinan Koho	1994-2014

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	JP 2010-134906 A (Seiko Epson Corp.), 17 June 2010 (17.06.2010), paragraphs [0061] to [0066], [0075] & US 2010/0100606 A1	1, 2, 11, 12 3-10, 13-20
Y	JP 2006-135982 A (Mitsubishi Electric Corp.), 25 May 2006 (25.05.2006), paragraphs [0114] to [0117], [0201] & US 2006/0164550 A1 & EP 1617333 A1 & WO 2004/095293 A1 & CN 1698042 A	3-10, 13-20
Y	JP 2000-500376 A (Creator Ltd.), 18 January 2000 (18.01.2000), pages 71 to 74 & US 5752880 A & EP 961645 A & WO 1997/018871 A2 & AU 2612097 A & CA 2237812 A	5-10, 15-20

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
25 February, 2014 (25.02.14)

Date of mailing of the international search report
04 March, 2014 (04.03.14)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. G06F13/00(2006.01)i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（I P C））

Int.Cl. G06F13/00

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1 9 2 2 - 1 9 9 6 年
日本国公開実用新案公報	1 9 7 1 - 2 0 1 4 年
日本国実用新案登録公報	1 9 9 6 - 2 0 1 4 年
日本国登録実用新案公報	1 9 9 4 - 2 0 1 4 年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
X Y	JP 2010-134906 A（セイコーエプソン株式会社）2010.06.17, 段落【0061】-【0066】、【0075】 & US 2010/0100606 A1	1, 2, 11, 12 3-10, 13-20
Y	JP 2006-135982 A（三菱電機株式会社）2006.05.25, 段落【0114】-【0117】、【0201】 & US 2006/0164550 A1 & EP 1617333 A1 & WO 2004/095293 A1 & CN 1698042 A	3-10, 13-20
Y	JP 2000-500376 A（クリエーター リミテッド）2000.01.18, 第71頁-第74頁 & US 5752880 A & EP 961645 A & WO 1997/018871 A2 & AU 2612097 A & CA 2237812 A	5-10, 15-20

☐ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの

「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）

「O」口頭による開示、使用、展示等に言及する文献

「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」同一パテントファミリー文献

国際調査を完了した日

2 5 . 0 2 . 2 0 1 4

国際調査報告の発送日

0 4 . 0 3 . 2 0 1 4

国際調査機関の名称及びあて先

日本国特許庁（I S A / J P）

郵便番号100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官（権限のある職員）

木村 雅也

電話番号 03-3581-1101 内線 3568

5 T

3 9 8 0